



**EPS**

Escola Politècnica

Superior

**Projecte/Treball Fi de Carrera**

**Estudi:** Eng. Tècn. Informàtica de Gestió. Pla 2001

**Títol:** Desenvolupament d'una aplicació mòbil tipus gimcana per plataformes Android

**Document:** Memòria

**Alumne:** Mariona Batalla Taylor

**Director:** Marta Fairén González

**Tutor:** Gustavo Ariel Patow

**Departament:** Informàtica i Matemàtica Aplicada

**Àrea:** LSI

**Convocatòria (mes/any):** 09/2014

**Índex**

1. Introducció, motivacions, propòsits i objectius del projecte	4
a. Propòsit i objectius	
b. Gimcanes	
c. Proves o punts d'interès	
d. Motivació personal	
e. Estructura del document	
2. Estudi de viabilitat	7
a. Recursos necessaris per a desenvolupar el projecte	
b. Avaluació prèvia de costos	
i. Estudi de viabilitat tecnològica	
ii. Estudi de viabilitat econòmica	
1. Despesa de recursos humans	
2. Despesa de maquinària	
3. Despesa de programari	
4.	
3. Metodologia	11
4. Planificació	13
a. Pla de treball i tasques planificades	
i. Estudi i definició de la mecànica del joc	
ii. Estudi de la plataforma Android	
iii. Estudi del programari per a la creació dels elements visuals	
iv. Disseny i creació dels elements visuals	
v. Disseny i implementació dels algoritmes de generació de joc i generació de rutes	
vi. Disseny i implementació dels algoritmes per a descarregar el joc i carregar-lo al dispositiu	
vii. Disseny i implementació dels algoritmes d'emmagatzemar respostes	
viii. Disseny i implementació dels algoritmes d'enviament de dades	
ix. Disseny i implementació dels algoritmes de correcció i avaluació del joc	
x. Verificació dels algoritmes implementats	
xi. Finalització de la documentació	
b. Temps estimat	
5. Marc de treball i conceptes previstos	17
a. Marc de treball	
b. Conceptes previs	
6. Requisits del sistema	18
a. Requisits part web	
b. Requisits part Android	
c. Requisits no funcionals	
7. Estudis i decisions	21
a. Maquinari	

b. Android	
c. Programari, llenguatges de programació, frameworks i llibreries	
i. PHP	
ii. Sublime Text	
iii. Codeanywhere	
iv. jQuery	
v. XML	
vi. API de Google Maps	
vii. Servidor HTTP Apache	
viii. phpMyAdmin	
ix. Eclipse	
x. Android SDK	
xi. Illustrator	
8. Anàlisi i disseny del sistema	36
a. Anàlisi	
i. Parts de l'aplicació	
ii. Diagrames de casos d'ús	
b. Disseny del sistema	
i. Estructura de la Base de Dades	
ii. Diagrama de classes	
c. Diagrames d'activitat	
9. Implementació i proves	50
a. Interfície d'usuari	
b. Implementació	
10. Implantació i resultats	71
a. Protecció de dades	
11. Conclusions	72
12. Treball futur	73

## 1 Introducció, motivacions, propòsit i objectius del projecte

Des de sempre, els jocs en equip han ajudat a millorar l'esperit de cooperació de les persones amb qui es comparteixen objectius. Mitjançant aquests jocs es fomenta la participació activa dels membres dels grups, millorant les relacions personals, fomentant el treball col·laboratiu i promovent la cohesió social entre ells.

Les gimcanes o sèrie de jocs per equips, generalment a l'aire lliure, es coneixen en alguns països com jocs múltiples o de carreres d'obstacles en els que varis equips participen superant nombroses proves competitives de temps, preguntes, recerca, habilitat i agilitat per anar guanyant quelcom a canvi, ja sigui punts, algun objecte o pistes per a passar a un altre nivell i seguir superant altres proves fins arribar al final.

En aquest joc, s'estableix un temps determinat per a acabar amb totes les proves, establertes des del començament. La gimcana és molt utilitzada en festes infantils, reunions familiars i retrobaments, com ara convivències o campaments, ja que aquesta classe d'activitats integren a les persones i les ajuden a estar actives d'una forma lúdica i divertida.

El nombre de proves i el nombre de participants d'una gimcana no està establert, i dependrà del que decideixi el creador.



Figura 1: Membres d'un grup de la gimcana familiar del Poble Espanyol

### 1.1 Propòsit i objectius

L'objectiu del present projecte final de carrera és el desenvolupament d'una aplicació per mòbils amb plataforma Android que generi una gimcana, per a poder realitzar una gimcana més completa i propera a la nostra era actual; incorporant totes les funcionalitats que tenen els telèfons mòbils a les eines per a resoldre un joc de proves, i crear així un concepte més actual del que és una gimcana tradicional. La creació de l'aplicació pretén fomentar i establir un diàleg de saber col·laboratiu i d'aportació de coneixements entre els components del mateix grup, que potencia la implicació en la creació de la resposta al misteri o la realització de la prova.

Android és un sistema operatiu basat en el kernel de **Linux**, dissenyat inicialment per a telèfons mòbils. Tanmateix la seva riquesa funcional ha permès que s'estengui a dispositius com tablets, netbooks, televisions i altres. Per a la realització del projecte s'utilitzarà el **SDK** (Software Development Kit) d'Android, dins l'entorn de programació **Eclipse**, utilitzant el llenguatge **Java**, per al que seria la part “usuari”. La part “màster”, que és la creació del joc, serà realitzada en un servidor amb **HTML**, **CSS**, **PHP** i **Javascript**.

## 1.2 Gimcanes

La gimcana és una eina que permet potenciar els valors de treball en equip, esforç, superació, motivació, sentiment de pertinença, lideratge, entre d'altres, i és accessible a tothom. Per aquesta raó també tenen molt d'èxit les gimcanes corporatives on, mitjançant les proves, es treballa el desenvolupament professional a través de dinàmiques de motivació, creativitat, presa de decisions, confiança personal, i habilitat de comunicació i de treball en equip; on els participants consoliden els seus coneixements a través d'interessants jocs i experiències de grup.

## 1.3 Proves o punts d'interès

La gimcana té per objectiu realitzar totes les proves que s'estableixen en l'itinerari de la pròpia gimcana. La ruta o seqüència de proves serà diferent -però equilibrada- per a cada equip. La modalitat d'itinerari serà lineal o seqüencial, on l'ordre de pas pels controls de cada punt és fix i, per tan, té un recorregut prefixat. Els punts de control estaran marcats al mapa de l'aplicació i caldrà resoldre'ls en el menor temps possible, però a l'hora amb el màxim de correctesa o encerts possible.

## 1.4 Motivació personal

Des de sempre he participat activament en totes les gimcanes que he pogut. M'atreu el fet de poder unir forces, coneixements i experiències amb un grup de gent i lluitar per un objectiu comú. Sobretot em motiven molt les gimcanes temàtiques, on totes les proves segueixen un eix narratiu o expositiu comú; penso que tenen un afegit de qualitat i d'utilitat i, fins i tot, que poden ser una forma original de promocionar i amenitzar qualsevol esdeveniment o lloc turístic.

És per això que un dia se'm va ocórrer anar un pas més enllà en les gimcanes tradicionals i afegir-hi aquest atribut (la tecnologia), per tal d'engrescar més als usuaris i fer-ne una experiència d'usuari ben enriquidora, així com incorporar noves formes de resoldre una prova: foto, twit...

## 1.5 Estructura del document

Aquest document està estructurat en 13 capítols, que són els següents:

1. **Introducció, motivacions, propòsit i objectiu del projecte.** En aquest capítol s'explicarà el perquè del desenvolupament d'aquest projecte, quins són els objectius proposats i com s'ha d'organitzar el desenvolupament.
2. **Estudi de viabilitat.** En aquest capítol es justifiquen els paràmetres que fan possible el desenvolupament del projecte.
3. **Metodologia.** Aquest capítol conté una explicació de la metodologia utilitzada.
4. **Planificació.** En aquesta etapa es defineix l'estratègia seguida per arribar als objectius plantejats.
5. **Marc de treball i conceptes previs.** En aquest capítol es descriuen els diversos aspectes relacionats amb el desenvolupament general del projecte, que ajudaran a entendre millor els següents capítols
6. **Requisits del sistema.** En aquest capítol es defineixen els requisits del software, els quals recullen, a grans trets, els objectius de l'aplicació juntament amb les funcionalitats que es volen obtenir.
7. **Estudis i decisions.** Aquesta secció conté una descripció de les eines utilitzades i l'ús que se'ls hi ha donat, tant de llibreries, com de software.
8. **Anàlisi i disseny del sistema.** Aquest apartat proporciona una comprensió precisa de les necessitats del sistema. És a dir, s'encarrega de la investigació del problema a resoldre, però no es materialitza la solució. Es tradueixen requeriments anomenats en capítols anteriors a un llenguatge més formal.
9. **Implementació i proves.** En aquest capítol es dona a conèixer com s'ha construït l'aplicació, les classes i els mètodes implementats que resulten més significatius per a la comprensió del funcionament del joc.
10. **Resultats.** En aquest capítol es mostren proves d'execució de l'aplicació per tal de veure'n les funcionalitats finals.
11. **Conclusions.** En aquest apartat s'exposaran les conclusions extretes un cop finalitzat el projecte.
12. **Treball futur.** En aquesta secció s'exposa tot allò que es podria millorar en l'aplicació, o ampliar-la.
13. **Bibliografia.** Aquest capítol conté totes les referències utilitzades pel desenvolupament del projecte.

## 2 Estudi de viabilitat

### 2.1 Recursos necessaris per a desenvolupar el projecte

Per desenvolupar aquest projecte no es requereix d'una gran infraestructura.

Els dispositiu amb què s'ha treballat durant el transcurs d'aquest projecte és un ordinador portàtil amb un processador Intel Core i7 – 3630QM, a una freqüència de rellotge de 2.40GHz, amb 4,00 GB de RAM i un disc dur de 500GB; tot i que per la realització d'aquest projecte es podria haver utilitzat un ordinador amb capacitats tècnicament inferiors.

Com a sistema operatiu, el projecte ha estat realitzat amb un Microsoft Windows 8 de 64 bits, processador Intel (R) Core (TM) i7-3630QM CPU @ 2.40GHz.

En quan al software utilitzat, s'ha fet ús de:

- ◆ Sublime Text: editor de text
- ◆ Eclipse: entorn de desenvolupament, amb els plugins AndroidSDK i ADT (Android Development Tools)
- ◆ Adobe Illustrator: per als gràfics de l'aplicació Android

Cal remarcar que no s'ha utilitzat cap emulador de servidor web, ja que he treballat directament amb el que treballo habitualment. Es tracta d'un hosting Linux amb una base de dades MySQL.

Pel que fa a la part Android, s'ha utilitzat un emulador, que és el que ve inclòs amb el propi SDK. A la part final del projecte, hem testejat l'aplicació, però, sota un dispositiu real.

### 2.2 Avaluació prèvia de costos

Pel desenvolupament de l'aplicació és necessari fer un estudi de viabilitat, tenint en compte la viabilitat econòmica, la viabilitat tècnica i la viabilitat legal. En el cas d'aquest projecte no tindrem problemes legals, així doncs només es consideraran:

- ◆ Estudi de viabilitat tecnològica
- ◆ Estudi de viabilitat econòmica

#### 2.2.1 Estudi de viabilitat tecnològica

Referent a la viabilitat tecnològica es considera que la tecnologia de la que prèviament disposem ja és suficient per assolir l'objectiu de la realització de l'aplicació a realitzar.

Com a material, a part d'un editor de text (o un entorn de desenvolupament que permeti treballar amb HTML, CSS, PHP i JavaScript), es necessitarà un programari nou, l'entorn Eclipse que, en quant a costos, no en tindrà cap d'afegit ja que és un programari de software lliure i gratuït.

## 2.2.2 Estudi de viabilitat econòmica

L'anàlisi econòmic s'ha separat en dues parts, corresponents a la despesa de recursos humans i la despesa de la maquinària.

### 2.2.2.1 Despesa de recursos humans

Per tal de calcular les despeses de recursos humans, associem un perfil de treballador per a cadascuna de les tasques:

- ♦ Cost analista: 6,5€/hora
- ♦ Cost programador: 4,6€/hora
- ♦ Cost dissenyador : 6,5€/hora

Nota: els salaris s'han extret dels **salaris mínims** que figuren al **conveni estatal de les consultories**<sup>1</sup>, conveni pel qual es regeixen les empreses que proporcionen serveis d'informàtica, considerant una jornada de 40 hores setmanals, i un total de (salaris mínims):

- ♦ Analista: 1569,25€ al mes
- ♦ Programador: 1103,04€ al mes
- ♦ Dissenyador: 1569,25€ al mes

---

1

Conveni estatal de les consultories: és el conveni d'aplicació obligatòria a totes les empreses que ofereixin serveis d'informàtica. Aquest conveni va ser firmat el 16/12/2008 i les últimes taules salarials vigents són les del 2009. El conveni va finalitzar el 31/12/2009, però per l'article 4 del mateix, es prorroga automàticament, fins que no es firmin unes noves taules salarials.



TASCA	PERFIL	HORES	COST
Estudi d'eines bàsiques web (html, css, javascript)	Analista	20	130,00 €
Estudi d'eines de programació amb crides a base de dades (php)	Analista	25	162,50 €
Estudi de llibreries	Analista	30	195,00 €
Estudi d'eines de programació per Android (java)	Analista	35	227,50 €
Disseny d'algoritme	Analista	100	650,00 €
Implementació de codi	Programador	380	1.746,48 €
Disseny web	Dissenyador	50	325,00 €
Disseny gràfic	Dissenyador	20	130,00 €
Proves i optimització	Programador	40	183,84 €
Memòria i manual d'usuari	Analista	120	780,00 €
<b>TOTAL</b>		<b>820</b>	<b>4.530,32 €</b>

### 2.2.2.2 Despesa de maquinària

En aquest cas no s'ha tingut cap despesa de hardware, ja que des d'un principi es disposaven dels components necessaris i no s'ha hagut de comptabilitzar cap cost addicional. Tot i així, s'ha comptabilitzat l'amortització de hardware:

$$Amortització = \frac{Cost\ Hardware}{Temps\ d'\ amortització} \cdot Temps\ de\ projecte$$

Considerant:

-Cost Hardware	900€
-Temps d'amortització	2 anys
-Temps de projecte	1 any

**S'obté una amortització de 450€.**

Per tant, sumem l'amortització a la despesa d'hores de personal i el cost total seria de  $4530,32€ + 450€ = 4980,32€$

### 2.2.2.3 Despesa de programari

A més, cal tenir en compte que s'utilitza un programari no gratuït, per a la realització dels elements gràfics, com és l'Adobe Illustrator, que té un preu actual de 839,88€ + IVA anuals, considerant que és el programari complet per una empresa.

Així doncs, el cost total seria 4980,32€ + 839,88€ = **5820,20€**

### 3 Metodologia

Abans de començar a dur a terme el projecte, es va consultar diferents metodologies per a trobar la que més s'adaptava al tipus de projecte que s'ha elaborat. S'ha optat per una metodologia que reforça a cada pas el treball fet anteriorment i consta dels següents passos:

1. Escollir la feina a desenvolupar.
2. Decidir quines eines de programació s'utilitzaran.
3. Aprendre el funcionament de les eines i llenguatges de programació escollits.
4. Estructurar la feina en parts segons les funcions que s'hagin de realitzar.
5. Desenvolupar la part corresponent seguint l'ordre d'estructura del treball.
6. Fer comprovacions al finalitzar cada part, per confirmar que el funcionament és correcte.
  - Si, al realitzar les comprovacions, el resultat no és el desitjat, es tornarà al punt 5 per realitzar els canvis oportuns a l'última part desenvolupada o a les anteriors, en el cas que fos necessari.
  - Si, al realitzar les comprovacions, el resultat és el desitjat, es desenvoluparà la següent part tornant al punt 5. Una vegada s'hagin finalitzat totes les parts, amb les seves respectives comprovacions, s'iniciarà el punt 7.
7. Unir totes les parts desenvolupades i comprovar que el funcionament és correcte.
  - Si, al realitzar les comprovacions, el resultat no és l'esperat, es tornarà a punt 5 per realitzar els canvis oportuns a l'última part desenvolupada o a les anteriors, en el cas que fos necessari
  - Si, al realitzar les comprovacions, el resultat no és l'esperat, s'iniciarà el punt 8.
8. Generar diferents models d'exemple per comprovar que el funcionament és l'esperat.
  - ◆ Si, al realitzar les comprovacions, el resultat no és el desitjat, es tornarà al punt 5 per realitzar els canvis oportuns a l'última part o a les anteriors, si és convenient.
  - ◆ Si, al realitzar les comprovacions, el resultat és el desitjat, s'iniciarà el punt 9.
9. Arrodonir la documentació feta al llarg del projecte.

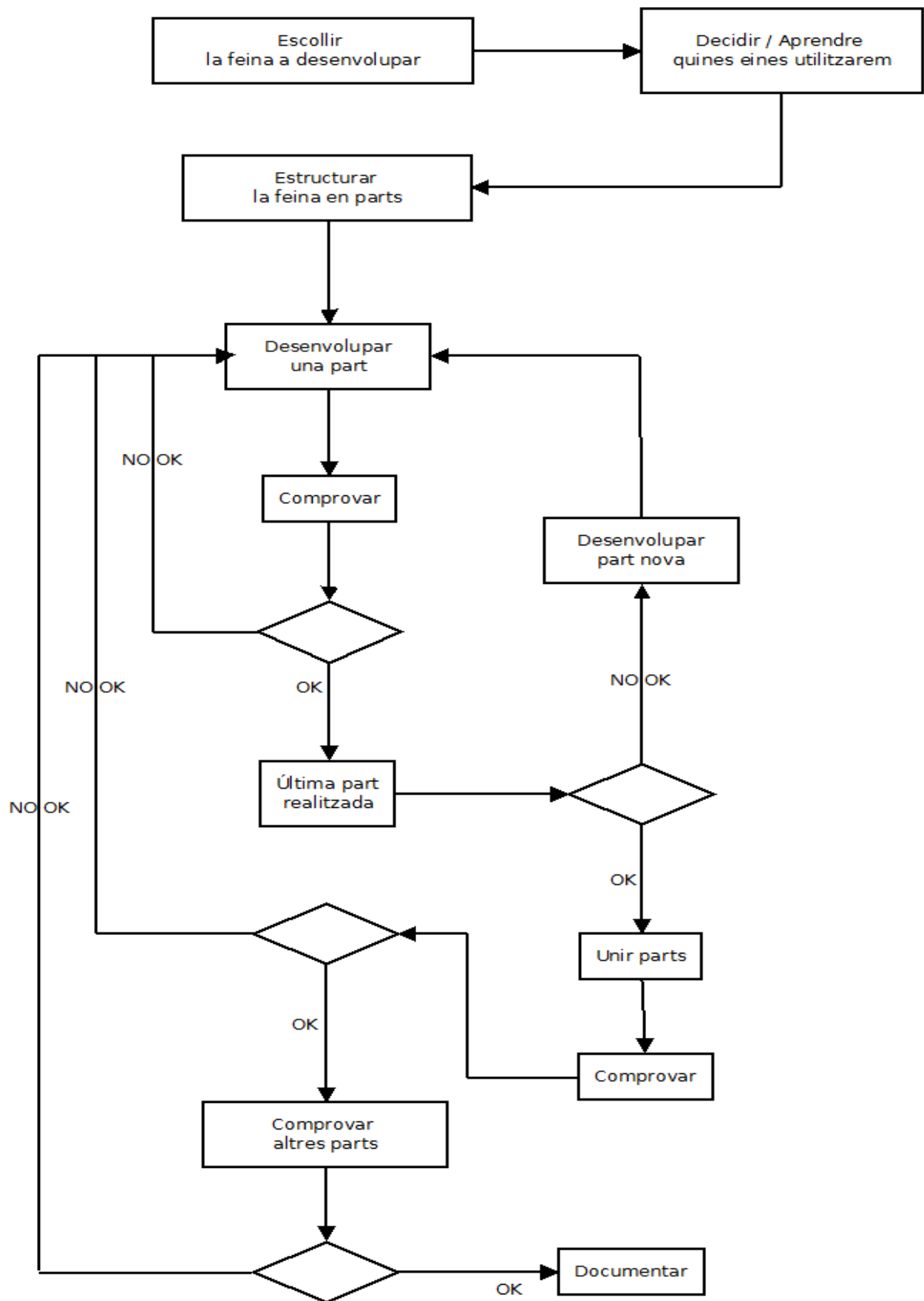


Figura2: Diagrama de flux de la metodologia que se seguirà

## 4 Planificació

Aquest projecte va néixer al juny del 2013, i estava planificat finalitzar-lo al juny del 2014.

Els mesos inicials van consistir en l'estudi de les diferents opcions que podrien ser d'utilitat per a assolir els objectius d'aquest projecte. Un cop decidit el sistema i la plataforma que es faria servir, va arribar l'etapa d'analitzar i estudiar el funcionament. Aquesta etapa, que va comprendre la recerca, estudi i aprenentatge de la nova plataforma, també va alternar-se amb l'etapa de realització de codi.

Un cop acabat el codi, verificat el funcionament, corregits els errors i millorat tot allò que fos possible millorar, es va procedir a “vestir” el projecte amb alguns elements bàsics gràficament més agradables.

### 4.1 Pla de treball i tasques planificades

Per realitzar la planificació del projecte, diferenciarem les següents tasques:

- ♦ Estudi i definició de la mecànica
- ♦ Estudi de la plataforma Android
- ♦ Estudi del programari per a la creació dels elements visuals
- ♦ Disseny i creació dels elements visuals
- ♦ Disseny i implementació dels algoritmes de generació de joc i generació de rutes
- ♦ Disseny i implementació dels algoritmes per a descarregar un joc i carregar-lo al dispositiu
- ♦ Disseny i implementació dels algoritmes d'emmagatzemar respostes
- ♦ Disseny i implementació dels algoritmes d'enviament de respostes
- ♦ Disseny i implementació dels algoritmes de correcció i avaluació del joc
- ♦ Verificació i proves dels algoritmes implementats
- ♦ Finalització de la documentació.

#### 4.1.1 Estudi i definició de la mecànica del joc

La primera tasca consistirà en analitzar la mecànica del joc, definint i especificant quins elements necessitem per a crear la infraestructura i per a utilitzar-lo com a usuari final.

#### 4.1.2 Estudi de la plataforma Android

En la següent tasca es comença a estudiar la plataforma Android, realitzant petits algoritmes inicials i tutorials per tal de familiaritzar-se amb la plataforma i assolir conceptes.

#### **4.1.3 Estudi del programari per a la creació dels elements visuals**

Aquesta fase, paral·lela a les anteriors, inclou els tutorials i estudis necessaris per a la realització i creació dels elements de disseny gràfic que s'han utilitzat per tal de fer més atractiva l'aplicació.

#### **4.1.4 Disseny i creació dels elements visuals**

Creació dels elements visuals gràfics per a l'aplicació.

#### **4.1.5 Disseny i implementació dels algoritmes de generació de joc i generació de rutes**

En aquesta tasca es pensaran i implementaran els algoritmes que generin un joc i les diferents rutes equitatives per a un mateix joc.

#### **4.1.6 Disseny i implementació dels algoritmes per a descarregar el joc i carregar-lo al dispositiu**

Aquí es crearan els algoritmes que permetin descarregar un joc del servidor i carregar-lo al dispositiu per tal que l'usuari pugui començar a jugar.

#### **4.1.7 Disseny i implementació dels algoritmes d'emmagatzemar respostes**

Aquesta fase consistirà en implementar els algoritmes que permetin als usuaris donar les seves respostes i a l'aplicació emmagatzemar-les.

#### **4.1.8 Disseny i implementació dels algoritmes d'enviament de dades**

En aquest apartat es duran a terme els algoritmes que permetin que l'aplicació, amb les respostes que ha emmagatzemat de l'usuari, envii aquestes dades al servidor de nou.

#### **4.1.9 Disseny i implementació dels algoritmes de correcció i avaluació del joc**

Aquí es crearan els algoritmes de correcció de les respostes enviades pels usuaris, i la posterior avaluació del joc, generant una puntuació.

#### **4.1.10 Verificació dels algoritmes implementats**

L'última de les tasques de desenvolupament serà la de comprovar que els algorismes generats facin les funcions per les que han estat creats.

#### **4.1.11 Finalització de la documentació**

La documentació és una tasca que s'ha de portar a terme constantment durant el desenvolupament del projecte.

### **4.2 Temps estimat**

Inicialment, es va plantejar que el projecte tindria una durada d'uns 10 mesos, per tant que estaria acabat aproximadament pel Juliol de 2014. A continuació mostrem la representació gràfica, mitjançant un diagrama de Gantt, d'aquest plantejament inicial:

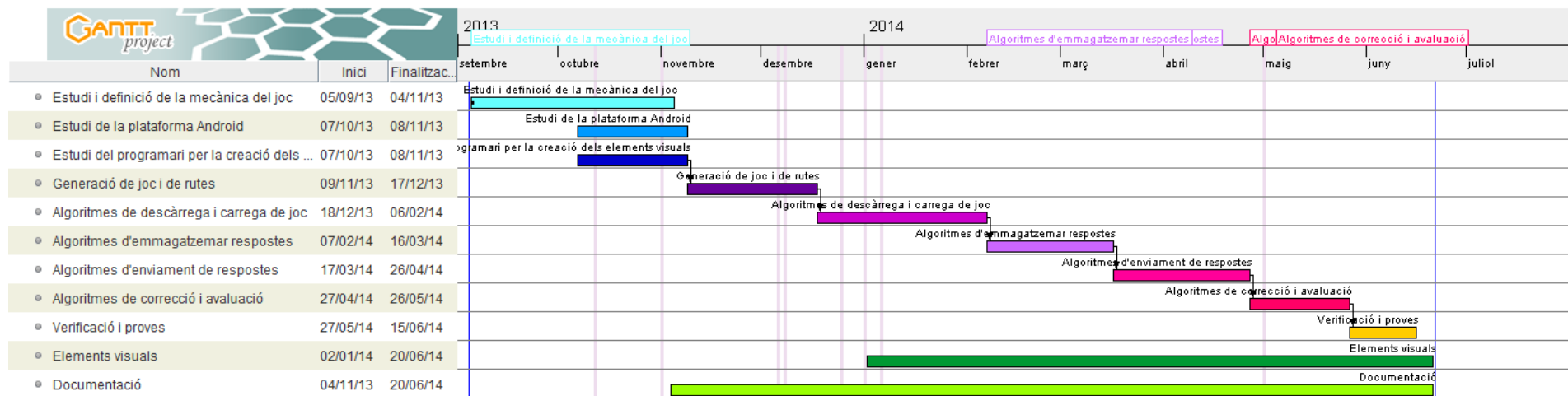


Figura: Diagrama de Gantt previst



## 5 Marc de treball i conceptes previs

En aquest apartat es comentaran tots els aplicatius referents al marc de treball i els conceptes necessaris per a una millor comprensió i seguiment del projecte.

### 5.1 Marc de treball

Una gimcana organitzada requereix de preparació prèvia, mai és quelcom espontani que es decideix al mateix moment. Prèviament s'ha pensat i decidit les proves, la manera de saber quina és la següent prova i com s'avaluaran les proves.

Tenim, per tant, l'equip que crea aquesta gimcana i pren les decisions següents:

- ◆ Modalitat de competició, que pot ser individual o per equips.
- ◆ Nombre de proves i localització d'aquestes.
- ◆ En què consisteix cada prova
- ◆ Sistema de puntuació de la prova
- ◆ Com es passa a la següent prova
- ◆ Quin és el premi final

Pel que fa als jugadors, ja siguin individuals o per equips, se'ls sol donar un paper per anar apuntant les puntuacions obtingudes en cada prova. Se'ls informa de la prova inicial que han de realitzar i a partir d'aquí, després de cada prova se'ls indica quina és la següent prova a realitzar.

Un cop finalitzades les proves, els jugadors han de tornar a la base inicial, on entregaran el paper de puntuacions i on es realitzarà la comprovació i la validesa de les proves. El resultat dels guanyadors pot determinar-se al moment, o bé al cap d'uns dies, segons la complexitat de l'avaluació de les proves.

### 5.2 Conceptes previs

Hem escollit la programació mitjançant el llenguatge de programació Java, ja que en la programació de les aplicacions per la plataforma Android s'utilitza, a través de l'SDK (Software Development Kit), i també és un llenguatge que s'ha treballat durant la carrera, dins l'entorn de programació Eclipse, que també coneixem i que és el recomanat per a treballar amb Android.

Així doncs, es va començar a cercar exemples de codi d'aplicacions per a Android, al mateix temps que s'estudiava l'arquitectura de les aplicacions, i es començava a desenvolupar aplicacions simples amb alguna de les funcionalitats que més endavant es requeririen.

Cal dir que aquest projecte es divideix en dues parts ben diferenciades:

- ◆ Part web: on es donen d'alta els usuaris, jocs i proves i des d'on es creen els diferents recorreguts per a cada joc
- ◆ Part aplicació Android: des d'on l'usuari es descarrega el joc, hi interactua, llegeix les proves i envia les respostes

## 6 Requisits del sistema

Per a poder establir els requisits que havia de complir l'aplicació es van dur a terme alguns anàlisis en diferents fases, per tal d'anar incorporant el que es necessitava, essent cada cop més específic i, a mesura que anava avançant la codificació anar-se adaptant als nous conceptes.

Podem distingir entre la part web i la part Android:

### 6.1 Requisits part web

Els requisits de la part web són els següents:

- ♦ Entrada a l'aplicació web a partir d'una pantalla d'identificació des d'on es demana el login i la contrasenya. No es podrà accedir a cap pantalla si no s'ha introduït una identificació correcta.
- ♦ Gestió dels usuaris. Cada usuari que accedeix mitjançant l'aplicació web és un màster, és a dir, podrà crear jocs. Aquest rol permetrà al màster donar d'alta, modificar, i/o eliminar jocs, jugadors i proves.
- ♦ Quan un usuari entra a l'aplicació podrà visualitzar els jocs, jugadors i proves que ha donat d'alta.
- ♦ Hi haurà una pantalla d'inici on apareixeran els jocs, jugadors i proves creats; així com també la possibilitat de crear-ne de nous, afegir proves a un joc donat i assignar jugadors a jocs.
- ♦ A la pantalla de creació de Jocs, Proves i Jugadors s'hi podrà accedir des de la pantalla d'inici de l'usuari màster. Destaquem les característiques i opcions dels mateixos:
  - **Joc:**
    - Nom del joc: serà el nom que s'especifica per la gimcana, que haurà de ser únic també en la base de dades. Serà un camp no modificable.
    - Localització: l'indret per on es durà a terme el joc, pot ser una població o una localització més acotada tipus un parc o una zona concreta.
    - Imatge del joc: serà la imatge que s'utilitzarà com a avatar del joc. No és obligatòria.
    - Base del joc: serà la localització on té lloc l'inici del joc, que també serà on s'hauran de dirigir els jugadors un cop finalitzat.
    - Data i hora d'inici: data i hora a partir de la qual el joc estarà disponible per a que els usuaris juguin.
    - Esquema cromàtic del joc: l'esquema de colors que es farà servir a l'aplicació Android, per tal de personalitzar-ho més.
    - Opció d'afegir una o més proves a un joc concret.
    - Donar d'alta el joc
    - Canviar l'estat d'un joc: el joc es crea per defecte amb l'estat tancat. L'usuari màster des d'aquí pot obrir el joc, per tal que els usuaris el tinguin disponible.
    - Cancel·lar

- **Jugador:**

- Nick: serà el nickname que identificarà al jugador en el sistema, serà únic i també serà un camp clau.
  - Password: la contrasenya d'accés de l'usuari jugador per tal d'accedir als jocs que estigui donat d'alta.
  - Nom: nom real de l'usuari jugador
  - Primer cognom: primer cognom real de l'usuari jugador
  - Segon cognom: segon cognom real de l'usuari jugador, que no és imprescindible.
  - Correu electrònic: e-mail de l'usuari jugador per tal de rebre informació.
  - Data de naixement: data de naixement de l'usuari jugador
  - Avatar: imatge pròpia del jugador que s'associarà a ell mateix, és opcional.
  - Opció d'assignar un jugador a un o més jocs concrets.
  - Donar d'alta el jugador.
  - Cancel·lar
- **Prova**
    - Prèviament s'haurà seleccionat el joc al qual pertany aquesta prova
    - Nom: serà el nom que identifiqui la prova d'aquest joc
    - Localització: serà la localització, adreça completa, d'on es trobarà la prova.
    - Enunciat: Breu explicació i contextualització – en el cas que sigui necessari- i enunciat o pregunta de la prova
    - Puntuació: nombre de punts que valdrà la prova
    - Tipus: de quin tipus serà la prova, hi ha els següents tipus:
      - ♦ Pregunta ABC: la resposta de la prova serà elegir la resposta correcta d'entre quatre respostes disponibles
      - ♦ Foto: la prova consistirà en carregar una foto amb els requisits que es demanen a l'enunciat.
      - ♦ Pregunta oberta: la prova consistirà en contestar una pregunta oberta, és a dir, no hi ha les possibles respostes disponibles.
        - Respostes ABC: només en el cas que el tipus de prova sigui “Pregunta ABC”, s'activarà el formulari de respostes, on es demanaran les quatre opcions de resposta diferents, i la resposta correcta.
        - Respostes correctes: només en el cas que el tipus de prova sigui “Pregunta oberta”, es demanaran quatre possibles respostes de la pregunta, per facilitar la tasca de correcció automàtica.
      - Donar d'alta prova.
      - Cancel·lar.
    - ♦ A partir d'un joc seleccionat, i tenint en compte el nombre de jugadors assignats al joc, generar els arxius per a cada jugador, per tal de tenir el seu joc “personalitzat”.

## 6.2 Requisits part Android

Els requisits de la part mòbil (Android) són els següents:

- ♦ Entrada a l'aplicació Android a partir d'una pantalla d'identificació des d'on es demana el login i la contrasenya. No es podrà accedir a cap pantalla si no s'ha introduït una identificació correcta.
- ♦ Un cop entra l'usuari, se li mostren els jocs dels quals està donat d'alta, i l'estat en què es troben aquests jocs.
- ♦ L'usuari podrà seleccionar el joc l'estat del qual sigui obert, per a poder entrar-hi i veure la informació del joc.
- ♦ L'usuari podrà començar a jugar si el joc ja està disponible per a interactuar-hi.
- ♦ Un cop l'usuari entri dins el joc, començarà la seqüència de pantalles de preguntes on l'usuari llegirà l'enunciat i haurà d'entrar la seva resposta.
- ♦ Un cop finalitzi l'última pregunta, se li demanarà a l'usuari que torni a la base i que envii les respostes.

### 6.3 Requisits no funcionals

Per tal que l'aplicació funcioni sense problemes, es requereix el següent software:

- ♦ Windows, qualsevol versió que permeti la instal·lació de l'entorn Eclipse.
- ♦ Servidor web amb connexió permanent, amb apache, MySql
- ♦ Eclipse
- ♦ Android SDK
- ♦ Sistema operatiu Android, versió mínima Android 2.2 (Froyo)

Pel que fa al hardware, l'utilitzat per a la implementació i proves és el següent:

- ♦ Toshiba Satellite
- ♦ Intel (R) Core (TM) i7-3630QM CPU @ 2.40GHz
- ♦ Dispositiu Android amb versió mínima Android 2.2 (Froyo)

## 7 Estudis i decisions

Aquí es presentaran les eines bàsiques que han permès la implementació tant de la part web, com de la part Android, així com també les eines que han servit de suport.

### 7.1 Maquinari

Per a la realització d'aquest projecte s'ha utilitzat un únic ordinador portàtil, amb les següents característiques:

- ◆ Processador: Intel (R) Core (TM) i7-3630QM CPU @ 2.40GHz
- ◆ Sistema operatiu: Windows 8, de 64 bits, processador basat en x64
- ◆ Memòria RAM: 4,00GB
- ◆ Disc dur: 500GB
- ◆ Targeta gràfica: AMD Radeon HD 7670M

Tot i que, com s'ha expressat amb anterioritat, també s'hauria pogut dur a terme amb un ordinador amb requisits inferiors.

### 7.2 Android

#### 7.2.1 Introducció

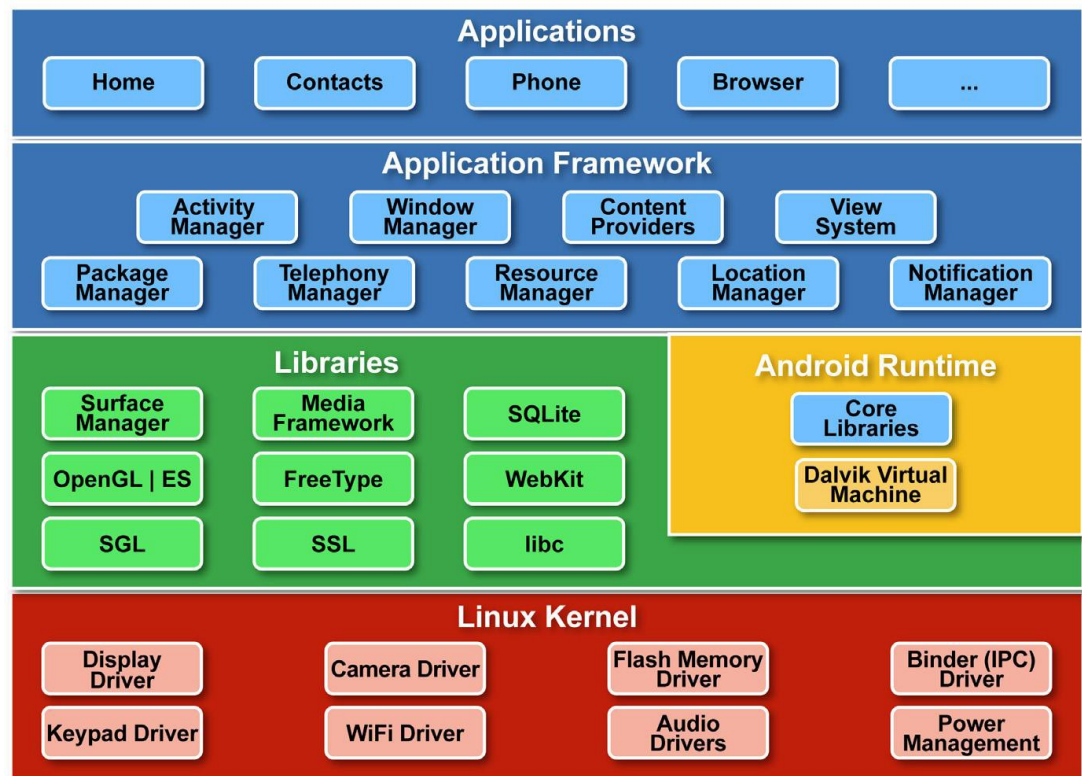


Android és un sistema operatiu basat en el kernel de Linux dissenyat principalment per a dispositius mòbils amb pantalla tàctil, com telèfons intel·ligents o tablets, i també per a rellotges intel·ligents, televisors i automòbils. Android va ser inicialment desenvolupat per Android, Inc. Tot i que Google va recolzar el projecte econòmicament i més tard va comprar l'empresa.

Android proporciona un framework molt ric que permet construir aplicacions i jocs innovadors per dispositius mòbils en un entorn de desenvolupament Java.

### 7.2.1.1 Capes d'Android

L'arquitectura d'Android està formada per diverses capes, la finalitat de la qual és facilitar el desenvolupament de les aplicacions, proporcionant els mitjans necessaris per a que el desenvolupador no hagi de programar a nivell de components físics. L'arquitectura segueix un patró de pila (stack), fet que permet que cada una d'aquestes capes utilitzi els serveis oferts per les anteriors, i que cada capa ofereixi els seus propis serveis a les capes de nivells superiors. Les capes són les que es mostren a l'esquema de la figura següent que detallem a continuació.



### 7.2.1.2 Aplicacions

Aquest nivell conté les aplicacions base, que inclouen un client de correu electrònic, un programa de SMS, un calendari, mapes, navegador, contactes i altres. A part de les bàsiques, també inclou totes aquelles aplicacions que l'usuari vagi afegint posteriorment, ja sigui de terceres empreses o no. Totes les aplicacions utilitzen els serveis, les API i les llibreries dels nivells anteriors.

### 7.2.1.3 Framework d'aplicacions

Representa fonamentalment el conjunt d'eines de desenvolupament de qualsevol aplicació. Tota aplicació que es desenvolupi per Android, ja siguin les pròpies del dispositiu, les desenvolupades per Google o terceres companyies, utilitzen el mateix conjunt de API i el mateix framework, representat per aquest nivell. Entre les API més importants es poden trobar les següents:

- ◆ **Activity Manager:** Conjunt de API que gestiona el cicle de la vida de les aplicacions en Android.
- ◆ **Window Manager:** Gestiona les finestres des de les aplicacions i utilitza la llibreria Surface Manager.
- ◆ **Telephone Manager:** Inclou totes les API vinculades a les funcionalitats pròpies del telèfon (trucades, missatges, etc.)
- ◆ **Content Provider:** Permet que qualsevol aplicació comparteixi les dades amb les demés aplicacions per Android. Per exemple, gràcies a aquesta API la informació de contactes, agenda, missatges, etc. serà accessible per altres aplicacions.
- ◆ **View System:** Proporciona un gran nombre d'elements per a poder construir interfícies d'usuari (GUI), com llistes, botons, “check-boxes”, mida de la finestra, etc. Inclou també algunes vistes estàndard per a les funcionalitats més freqüents.
- ◆ **Location Manager:** Possibilita a les aplicacions l'obtenció d'informació de localització i posicionament.
- ◆ **Notification Manager:** Mitjançant el Notification Manager les aplicacions, utilitzant un mateix format, comuniquen a l'usuari events que succeeixin durant l'execució: una trucada entrant, un missatge rebut, connexió Wi-Fi disponible, etc. Si duen alguna acció associada, en Android denominada Intent, (per exemple, atendre una trucada rebuda) aquesta s'activa mitjançant un clic.
- ◆ **XMPP Service:** Col·lecció d'API per utilitzar aquest protocol d'intercanvi de missatges basats en XML.

#### 7.2.1.4 Llibries

La següent capa inclou un grup de llibries de C/C++ utilitzades per diversos components del sistema, que proporcionen a Android la major part de les capacitats més característiques. Junt amb el kernel (nucli) basat en Linux, aquestes llibries constitueixen el cor d'Android.

Entre les llibries més importants, es poden trobar les següents.

- ◆ **Llibreria libc:** inclou totes les capçaleres i funcions segons l'estàndard de llenguatge C. Totes les altres llibries es defineixen en aquest llenguatge.
- ◆ **Llibreria Surface Manager:** és l'encarregada de compondre els diferents elements de navegació de pantalla. Gestiona també les finestres de les aplicacions actives en cada moment.

- ♦ **OpenGL/SGL i SGL:** representen les llibreries gràfiques i, per tant, sustenten la capacitat gràfica d'Android. OpenGL/SGL treballa amb gràfics en 3D i permet utilitzar, en cas que el dispositiu mòbil ho permeti, el hardware encarregat de proporcionar gràfics 3D. Per altra banda, SGL proporciona gràfics en 2D, fet pel qual serà la llibreria més utilitzada habitualment per la majoria d'aplicacions. Una característica important de la capacitat gràfica d'Android és que és possible desenvolupar aplicacions que combinin 3D i 2D.
- ♦ **Llibreria Media Libraries:** proporciona tots els còdecs necessaris pel contingut multimèdia suportat en Android (vídeo, àudio, imatges estàtiques i animades, etc.).
- ♦ **Free Type:** Permet treballar de forma ràpida i senzilla amb els diferents tipus de fonts.
- ♦ **Llibreria SSL:** possibilita la utilització del protocol SSL per establir comunicacions segures.
- ♦ **Llibreria SQLite:** creació i gestió de bases de dades relacionals.
- ♦ **Llibreria WebKit:** proporciona un motor per les aplicacions de tipus navegador i forma el nucli de l'actual navegador, inclòs per defecte a la plataforma Android.

#### 7.2.1.5 Entorn d'execució d'Android

Se situa al mateix nivell que les llibreries d'Android. Aquest entorn està format per les Core Libraries, que són llibreries amb motles classes de Java i la màquina virtual Dalvik.

#### 7.2.1.6 Kernel Linux

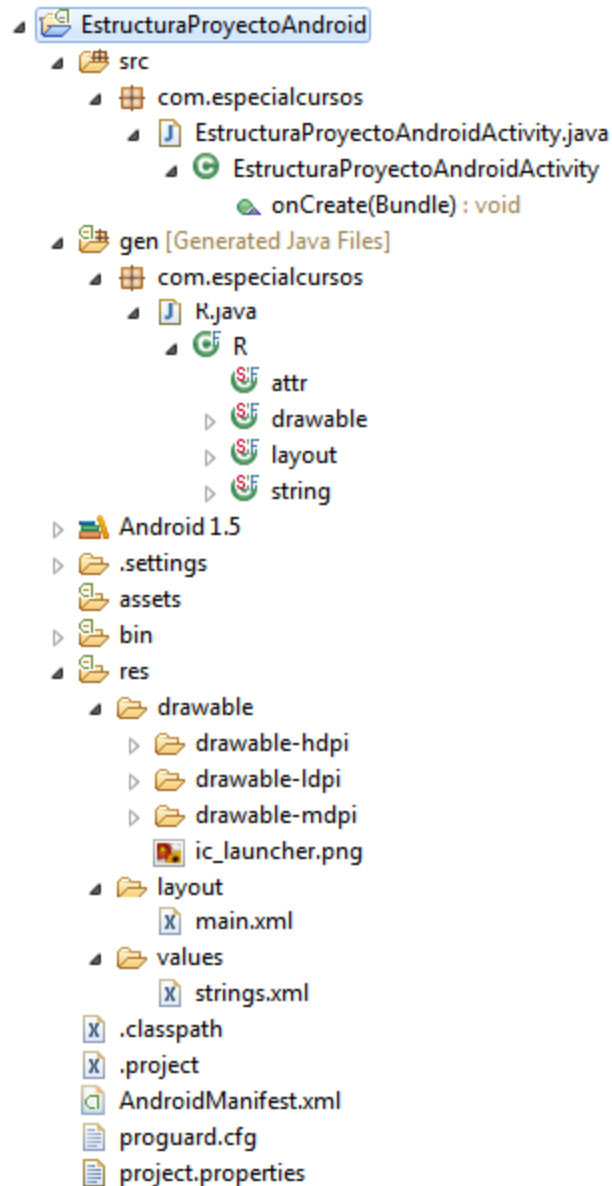
Android utilitza el nucli de Linux 2.6 com una capa d'abstracció pel hardware disponible als dispositius mòbils. Aquesta capa conté els drivers necessaris per a que qualsevol component hardware pugui ser utilitzat mitjançant les crides corresponents. Sempre que un fabricant inclogui un nou element de hardware, el primer que s'ha de fer per a que pugui ser controlat per a que pugui ser utilitzat des d'Android és crear les llibreries de control o drivers necessaris dins d'aquest kernel.

#### 7.2.2 Estructura de jeràrquica de carpetes d'una aplicació Android

L'entorn de desenvolupament crea una estructura de directoris bàsica que serà comuna en tots els projectes per aplicacions Android i que facilitarà l'organització dels diferents arxius que compondran l'aplicació.

L'estructura creada per defecte és la següent, i detallarem les més destacades:





- ♦ **/src:** és una de les carpetes principals del projecte. Conté tots els arxius de codi font Java que componen l'aplicació, entre elles les Activity, que inclouen els mètodes fonamentals del cicle de vida.
- ♦ **/gen:** Aquí hi trobarem l'arxiu **R.java**, que referencia tots els recursos del projecte. No s'ha de modificar mai.
- ♦ **/bin:** Conté els arxius compilats de l'aplicació.
- ♦ **/res:** Carpeta on resideixen tots els recursos accessibles des de la classe R, i on hi trobem les següents subcarpetes a destacar:
  - **drawable:** directori que conté tots els recursos gràfics de l'aplicació, organitzats en funció de la resolució de la pantalla que mostri l'aplicació.
  - **anim:** contindrà descriptors xml d'animacions
  - **layout:** directori on hi haurà els fitxers que defineixen la interfície d'usuari, els descriptors xml de les vistes.

- **values:** fitxers xml amb parelles de clau-valor, com ara Strings (conté tot el text que utilitza la aplicació) o colors, etc.

♦ **AndroidManifest.xml:** Aquest fitxer és imprescindible, descriu les característiques fonamentals de l'aplicació i defineix cada un dels components.

## 7.2.3 Components d'una aplicació Android

### 7.2.3.1 Activities

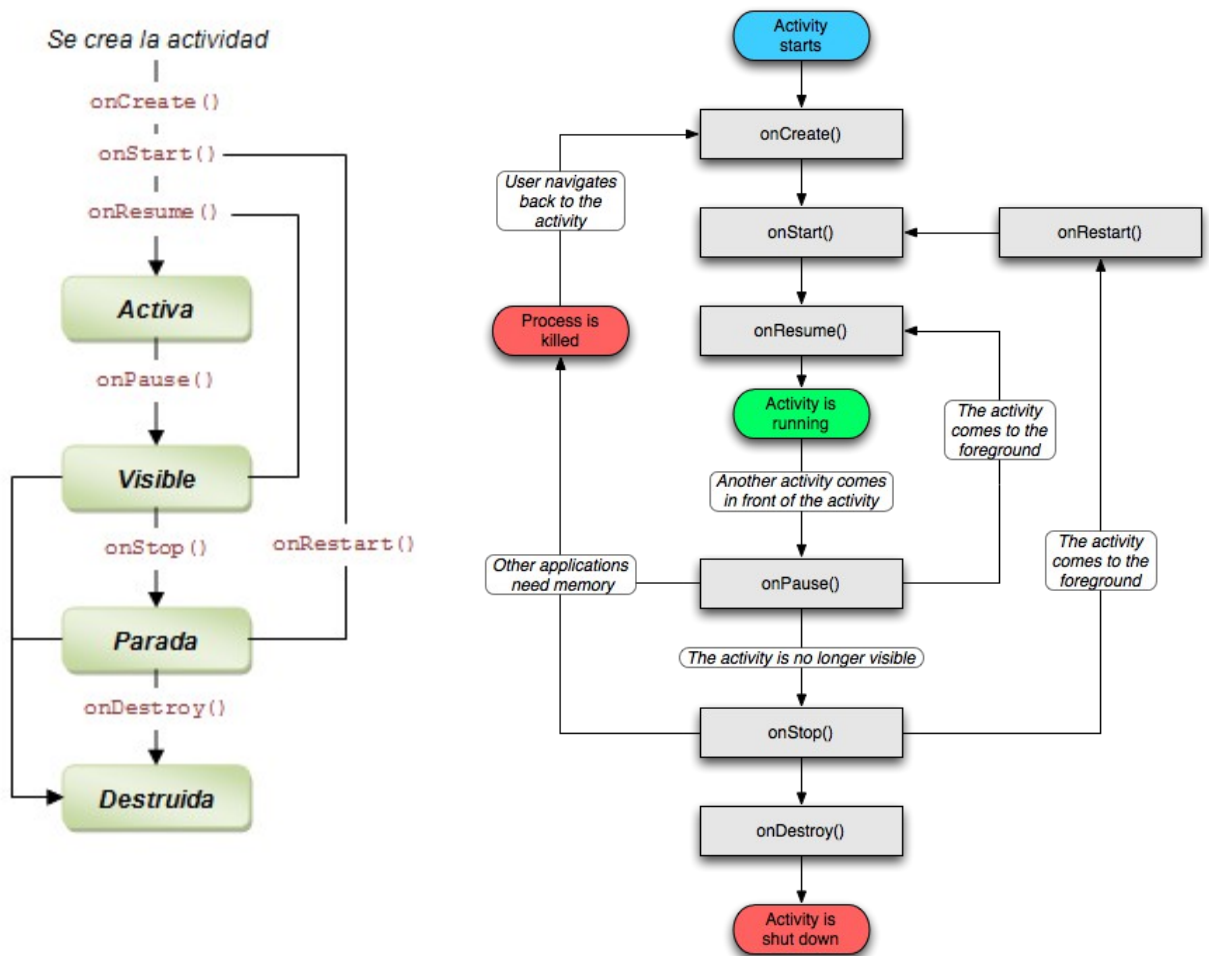
Les activitats (activities) representen el component principal de la interfície gràfica d'una aplicació Android. Es pot pensar com l'element anàleg a una finestra o pantalla en qualsevol altre llenguatge visual. És el component més habitual de les aplicacions per a Android, i se'n pot controlar completament el seu cicle de vida, que mostrem a continuació:

#### 7.2.3.1.1 Cicle de vida de les activitats

Les activitats són les que realment controlen el cicle de vida de les aplicacions, ja que l'usuari no canvia d'aplicació, sinó d'activitat. El sistema manté una pila amb les activitats prèviament visualitzades. Les activitats tenen quatre possibles estats:

- ♦ **Activa (Running):** L'activitat està a sobre de la pila, per tant és visible i té l'enfocament.
- ♦ **Visible (Paused):** L'activitat és visible però no té l'enfocament. Una aplicació passa a aquest estat quan activa una altra activitat amb alguna part transparent o que no ocupa tota la pantalla. Si una activitat passa a estar tapada completament, passa a estar parada.
- ♦ **Parada (Stopped):** Quan una activitat no és visible. En aquest cas, s'haurà de guardar l'estat de la interfície d'usuari, preferències, etc.
- ♦ **Destruïda (Destroyed):** Quan l'activitat acaba al cridar-se el mètode finish(), o és matada pel sistema.

Cada cop que una activitat canvia d'estat es generen uns events que poden ser capturats per mètodes de l'activitat. A continuació es mostren un parell d'esquemes que il·lustren el cicle de vida d'una activitat i els mètodes:



Detallem a continuació els mètodes que figuren a les imatges anteriors:

- ♦ **onCreate():** cridat quan l'activitat és creada. Aquí és on es creen les views, on s'enllacen les dades amb els components visuals, etc. Aquest mètode també proveeix amb un Bundle (permet passar dades entre activitats) que conté l'estat previ congelat de l'activitat, si n'hi ha un. Sempre està seguit de l'onStart().
- ♦ **OnStart():** es crida quan l'activitat esdevé visible per l'usuari. Seguit per onResume() si l'activitat ve al primer pla, o per onStop() si s'amaga.
- ♦ **OnResume():** cridada quan l'activitat comença a interactuar amb l'usuari. A aquest punt, l'activitat es troba al cim de la pila d'activitats. Sempre està seguit de onPause().
- ♦ **OnPause():** cridada quan el sistema és a punt de començar a reprendre una activitat anterior. S'utilitza normalment per a confirmar canvis no guardats en dades persistents, per a parar animacions i altres accions que puguin estar consumint CPU, etc. Les implementacions d'aquest mètode han de ser molt ràpides perquè la pròxima activitat no serà represa fins que aquest mètode torni. Va seguit de onResume() si l'activitat retorna al primer pla, o onStop() si esdevé invisible per l'usuari.

- ♦ **OnStop():** cridat quan l'activitat ja no és visible per l'usuari, perquè una altra activitat està essent començada i l'està cobrint. Això pot passar o bé perquè una nova activitat ha començat, una activitat existent es porta al davant o bé està sent destruïda.
- ♦ **OnRestart():** Cridat després que l'activitat s'hagi aturat, abans de posar-se en marxa de nou. Sempre seguit pel mètode onStart().
- ♦ **onDestroy():** L'última crida que es rep abans que una activitat es destrueixi. Això pot passar o bé perquè l'activitat està acabant (és a dir, s'ha cridat explícitament el mètode finish()), o bé perquè el sistema està destruint temporalment aquesta instància de l'activitat per a alliberar espai.

### 7.2.3.2 View

Les vistes (View) són els components bàsics amb els que es construeix la interfície gràfica de l'aplicació. Per exemple, un objecte de tipus View seria un botó, un quadre de text, una llista desplegable, imatges, etc.

### 7.2.3.3 Intent

Un intent és un element bàsic de comunicació entre els diferents components d'Android. Són els missatges o peticions que són enviats entre els diversos components d'una aplicació o entre aplicacions diferents, consisteix bàsicament en la voluntat de realitzar alguna acció, generalment associada a unes dades. Llançant un Intent, una aplicació pot delegar la feina a una altra aplicació. Els Intents estan inclosos al fitxer AndroidManifest perquè descriuen on i quan pot començar una activitat.

### 7.2.3.4 Services

Un component Service representa una aplicació executada sense interfície d'usuari i que generalment s'executa en segon pla mentre altres aplicacions (aquestes amb interfície) són les que estan actives. Un exemple típic és el reproductor de música. Poden realitzar qualsevol tipus d'accions, com ara actualitzar dades, llençar notifikacions o fins i tot mostrar elements visuals, com podrien ser les activitats.

### 7.2.3.5 Content provider

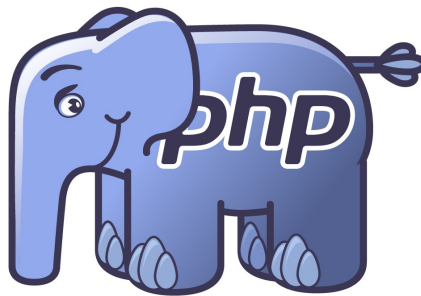
Amb el proveïdor de continguts (content provider) qualsevol aplicació d'Android pot emmagatzemar dades en un fitxer, en una base de dades SQLite o en qualsevol altre format que es consideri. A més aquestes dades poden ser compartides entre les diverses aplicacions. Una classe que implementi el component ContentProvider contindrà una sèrie de mètodes que permeten emmagatzemar, recuperar, actualitzar i compartir les dades d'una aplicació. És a dir, és el mecanisme que s'ha definit a Android per a compartir dades entre aplicacions.

### 7.2.3.6 Broadcast receiver

Un broadcast receiver és un component destinat a detectar i a reaccionar davant d'alguns missatges o events globals generats pel sistema (com per exemple: "bateria baixa", "SMS rebut", "Targeta SD insertada", etc.) o per altres aplicacions (anomenats Intents), és a dir, s'utilitza per llençar alguna execució dins de l'aplicació actual quan es produeixi un determinat event (generalment, obrir un component Activity). Aquest component no té una interfície d'usuari associada, però pot utilitzar l'API Notification Manager per avisar a l'usuari de l'event produït a través de la barra d'estat del dispositiu mòbil. Aquest complement s'implementa a través de la classe BroadcastReceiver.

## 7.3 Programari, llenguatges de programació, frameworks i llibries

### 7.3.1 PHP



PHP és un acrònim recursiu que significa PHP Hypertext Pre-processor. PHP és un llenguatge de programació interpretat, dissenyat originalment per a la creació de pàgines web dinàmiques amb accés a informació emmagatzemada en una base de dades. S'utilitza principalment per a la interpretació del costat del servidor, però actualment pot ser utilitzat des d'una interfície de línia de comandes o en la creació d'altres tipus de programes.

- ♦ És considerat un llenguatge fàcil d'aprendre, ja que en el desenvolupament es varen simplificar diferents especificacions.
- ♦ El codi fons que s'utilitza en PHP és invisible al navegador web i al client, ja que és el servidor el que s'encarrega d'executar el codi i enviar el resultat HTML al navegador.
- ♦ Té capacitat de connexió amb la majoria dels motors de base de dades que s'utilitzen a l'actualitat.
- ♦ Té capacitat per a expandir el seu potencial utilitzant mòduls (anomenats extensions).
- ♦ Posseeix una àmplia documentació al lloc web oficial.
- ♦ És lliure, pel que es presenta com una alternativa de fàcil accés per a tots.
- ♦ Permet aplicar tècniques de programació orientada a objectes.
- ♦ No requereix definició de tipus de variables, tot i que les variables es poden avaluar també pel tipus que s'estiguin utilitzant en temps d'execució.
- ♦ Fa ús d'excepcions.

### 7.3.2 Sublime Text



Sublime Text és un editor de text i editor de codi font creat en Python, desenvolupat originalment com una extensió de Vim que, amb el temps, va anar creant una identitat pròpia, tot i que encara conserva el mode d'edició tipus vi anomenat Vintage mode. Algunes de les característiques són:

- ♦ Té una previsualització de l'estructura del codi
- ♦ Suport per a 43 llenguatges de programació i text pla, entre els quals hi ha ASP, C, C#, C++, CSS, HTML, Java, JavaScript, Perl, PHP, Python, Rails, Ruby, SQL i XML.
- ♦ Remarcatge de sintaxis configurable
- ♦ Cerca dinàmica d'expressions regulats
- ♦ Autocompletat
- ♦ Marcat de claus que tanquen i obren un bloc
- ♦ Suport de Snippets i Plugins
- ♦ Té un intèrpret de Python amb el qual es poden realitzar multitud de tasques
- ♦ Facilita la lectura del codi ressaltant les expressions pròpies del llenguatge que s'està emprant.

### 7.3.3 Codeanywhere



Codeanywhere és un editor de codi via navegador amb un client de ftp integrat, i que suporta tots els formats populars de web (HTML, PHP, Javascript, CSS i XML). Permet codificar des de qualsevol lloc amb accés a la xarxa, només cal loguejar-se des de qualsevol ordinador o smartphone, i tots els servidors (FTP, SFTP i Dropbox) i els arxius estaran disponibles. Algunes de les característiques més destacades:

- ♦ Permet codificar des de pràcticament qualsevol lloc amb accés a la xarxa
- ♦ És ràpid
- ♦ Compatible amb la majoria de navegadors
- ♦ Existeix l'aplicació per Android i per iOS.

- ♦ No es requereixen instal·lacions
- ♦ Disposa de totes les opcions dels programes instal·lables: sintaxi ressaltada, codi plegable, mostra el número de la línia...
- ♦ Capacitat de connexió als servidors que es desitgin, no hi ha límit
- ♦ Té suport dropbox
- ♦ Els fitxers se sincronitzen automàticament

#### 7.3.4 jQuery



jQuery és una llibreria de Javascript, que permet simplificar el mode d'interactuar amb els documents HTML, manipular l'arbre DOM (Model d'Objectes del Document), fer ús d'esdeveniments, desenvolupar animacions i agregar interacció mitjançant tècniques AJAX a pàgines web. La característica principal de la llibreria és que permet canviar el contingut d'una pàgina web sense necessitat de recarregar-la, mitjançant la manipulació de l'arbre DOM i peticions AJAX. Per això s'utilitzen les funcions `$()` o `jQuery()`.

#### 7.3.5 XML



XML (Extensible Markup Language – llenguatge de marques extensible) és un llenguatge de marques desenvolupat pel World Wide Web Consortium (W3C), utilitzat per emmagatzemar dades de forma llegible. Deriva del llenguatge SGML i permet definir la gramàtica de llenguatges específics per estructurar grans documents. XML dóna suport a Bases de Dades, i és molt útil quan diverses aplicacions han d'establir comunicacions entre sí o integrar informació.

La tecnologia XML cerca donar una solució al problema d'expressió de la informació estructurada de manera més abstracta i reutilitzable possible. L'estructura de l'XML és un arbre amb trossos d'informació, anomenats elements, que són senyalats mitjançant etiquetes, que tenen la forma `<etiqueta></etiqueta>`.

Avantatges:

- ♦ És extensible: és possible estendre l'XML amb l'addició de noves etiquetes, de mode que es pugui continuar utilitzant sense cap complicació.

- ♦ L'analitzador és un component estàndard, no és necessari crear un analitzador específic per a cada versió del llenguatge XML. Això possibilita l'ús de qualsevol dels analitzadors disponibles. D'aquesta manera s'eviten bugs i s'accelera el desenvolupament de les aplicacions.
- ♦ Si un tercer decideix utilitzar un document creat en XML, és senzill entendre l'estructura i processar-la. Millora la compatibilitat entre aplicacions. Es poden comunicar aplicacions de diferents plataformes, sense importar l'origen de les dades.

### 7.3.6 API de Google Maps



Google Maps és un servei gratuït que ens ofereix Google amb mapes del món, fotos, rutes, ubicacions i altres característiques basades en mapes i localitzacions; fàcilment integrable a qualsevol aplicació o pàgina web.

L'API de Google Maps és un conjunt d'APIs de Google Maps que permeten, entre altres coses, crear aplicacions mòbils basades en ubicacions.

### 7.3.7 Servidor HTTP Apache



És un servidor web HTTP de codi obert, per plataformes Unix, Windows, Macintosh i altres, que implementa el protocol HTTP/1.1 i la noció del lloc virtual. Apache és el servidor HTTP més utilitzat. És utilitzat principalment per enviar pàgines web estàtiques i dinàmiques al World Wide Web.

Avantatges:

- ♦ Modular



- ♦ Codi obert
- ♦ Multiplataforma
- ♦ Extensible
- ♦ Popular, que implica que és més fàcil de trobar ajuda i suport.

### 7.3.8 PhpMyAdmin



phpMyAdmin és una eina online escrita en PHP que serveix per administrar bases de dades MySQL. Actualment pot crear i eliminar Bases de Dades, crear, eliminar i alterar taules, eliminar, editar i afegir camps, executar qualsevol sentència SQL, administrar claus en camps, administrar privilegis, exportar dades en varis formats i, a més a més, està disponible en molts idiomes. Cal destacar que es troba sota llicència GPL (General Purpose Licence – Llicència Pública General)

### 7.3.9 Eclipse



Eclipse és un programa compost per un conjunt d'eines de programació de codi obert multi-plataforma, que serveix per a desenvolupar aplicacions de qualsevol tipus. Aquesta plataforma típicament ha estat utilitzada per crear entorns de desenvolupament integrats (Integrated Development Environment, IDE), aplicacions que proporcionen eines per facilitar la codificació de software, com l'IDE de Java, el que s'ha fet servir, anomenat Java Development Toolkit (JDT) i el compilador que s'integra com part d'Eclipse.

L'entorn de desenvolupament integrat (IDE) utilitza mòduls (plug-ins) per a proporcionar a l'usuari les funcionalitats que es necessitin, i que permeten escriure qualsevol extensió desitjada.

S'ha utilitzat l'Eclipse per a codificar la part Android (Java), concretament la versió JUNO.

### 7.3.10 Android SDK



L'SDK (Software Development Kit) d'Android, inclou un conjunt d'eines de desenvolupament. Comprèn un depurador de codi, llibreries, un simulador de telèfons, documentació, exemples de codis i tutorials. La plataforma integral de desenvolupament (IDE) oficial és Eclipse, juntament amb el complement ADT (Android Development Tools plugin), per crear i depurar aplicacions. A més a més, es poden controlar dispositius Android que estiguin connectats.

Les actualitzacions de l'SDK estan coordinades amb el desenvolupament general d'Android. També suporta versions antigues d'Android, per si els programadors necessiten instal·lar aplicacions en dispositius antics o obsolets.

S'ha utilitzat, juntament amb Eclipse, per a codificar la part Android (java) del projecte.

### 7.3.11 Illustrator



Adobe Illustrator és un editor de gràfics vectorials en forma de taller d'art, que treballa sobre un tauler de dibuix, conegut com taula de treball. És desenvolupat i comercialitzat per Adobe Systems. Conté opcions creatives un accés més senzill a les eines i una gran versatilitat per a produir ràpidament gràfics flexibles utilitzats en maquetació i publicació d'elements, vídeo, publicacions web i dispositius mòbils. Disposa d'una gran quantitat d'eines i plugins. S'ha utilitzat pels gràfics de l'aplicació mòbil.

## 8 Anàlisi i disseny del sistema

### 8.1 Anàlisi

En aquest apartat s'explicaran les diferents parts que té l'aplicació i els diagrames de casos d'ús.

#### 8.1.1 Parts de l'aplicació

Un cop decidits els requisits i escollides les eines que s'utilitzaran, s'analitzaran les diferents parts que tindrà l'aplicació. L'aplicació, com s'ha dit anteriorment, tindrà dues parts ben diferenciades: la part web i la part Android.

- ♦ Part web: únicament per a màsters, servirà per gestionar els jocs/gimcanes.
- ♦ Part aplicació Android: únicament per jugadors (usuaris finals), servirà per a interactuar amb el joc.

A continuació es mostraran els diagrames de casos d'ús per a mostrar les funcionalitats del sistema, separats per cada part.

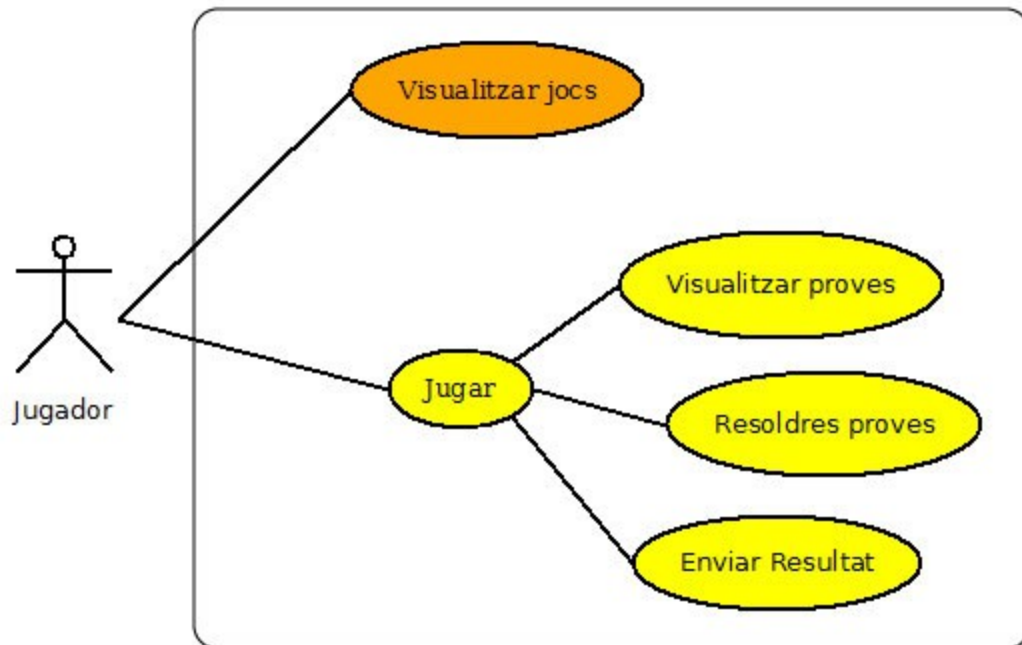
#### 8.1.2 Diagrames de casos d'ús

Com que el projecte consta de dues parts, clarament diferenciades, es mostraran dos diagrames de casos d'ús general: pel rol de màster (o administrador) i pel rol de jugador, que utilitzaran la part web i l'aplicació Android respectivament.

A continuació es mostra el diagrama de casos d'ús que fa referència a la part web de l'aplicació, aquella on només hi interactuarà el màster:



Seguidament es mostra el segon diagrama de cas d'ús, que correspon a la part Android, amb la que interactuarà l'usuari jugador.



Als diagrames anteriors es pot apreciar de forma molt general el funcionament de l'aplicació. A continuació s'explicarà breument cada una de les funcionalitats, per a aclarir conceptes.

Cal destacar, però, que qualsevol dels usuaris que vol accedir a l'aplicació primer ha de passar per una pantalla de "login", on haurà d'introduir el nom d'usuari i contrasenya. En cas contrari, no tindrà accés a l'aplicació.

## Part Web

### Administració de jocs

Aquí es podran realitzar les diferents opcions pel que fa als jocs:

- ♦ Crear joc: permetrà donar d'alta un joc a la base de dades
- ♦ Visualitzar jocs: mostrarà un llistat on apareixerà un resum dels jocs generats pel màster actual.
- ♦ Modificar joc: permetrà modificar atributs d'un joc ja creat

### Administració de proves

Aquí es podran realitzar les diferents opcions pel que fa a les proves:

- ♦ Crear prova: permetrà donar d'alta una prova -pertanyent a un joc- a la base de dades
- ♦ Visualitzar proves: mostrarà un llistat on apareixerà un resum de les proves generades pel màster actual.
- ♦ Modificar prova: permetrà modificar atributs d'una prova ja creada

## Administració de jugadors

Aquí es podran realitzar les diferents opcions pel que fa als jugadors:

- ♦ Crear jugador: permetrà donar d'alta un jugador a la base de dades
- ♦ Visualitzar jugadors: mostrarà un llistat on apareixerà un resum dels jugadors generats pel màster actual.
- ♦ Modificar jugador: permetrà modificar atributs d'un jugador ja creat

## Registrar-se al sistema

Des d'aquest mòdul, un màster podrà registrar-se al sistema per tal de poder començar a crear un joc.

## Generar fitxers

Permetrà, un cop acabat de configurar el joc, les proves i els jugadors que hi jugaran, generar els fitxers XML per a cada un dels jugadors.

## Correcció i validació joc

Un cop el joc hagi estat jugat, el màster podrà fer la correcció de la validesa de les proves que requereixin correcció manual (tipus foto, etc.).

## Part Android

### Visualitzar jocs

Un cop l'usuari entri al sistema, mostrarà tots els jocs on està donat d'alta l'usuari com a jugador, així com l'estat.

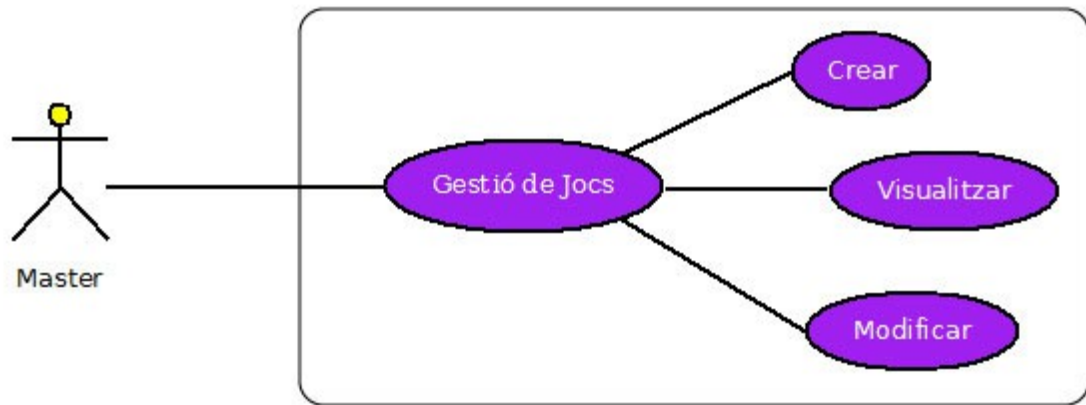
### Jugar

A l'entrar a un joc preparat per a ser jugat, el jugador podrà visualitzar una a una les proves per tal de llegir-ne l'enunciat i les especificacions, enviar una resposta a la prova i, quan acabi totes les proves, generar el fitxer de respostes al servidor.

### 8.1.3 Diagrames i fitxes de cas d'ús de la part web

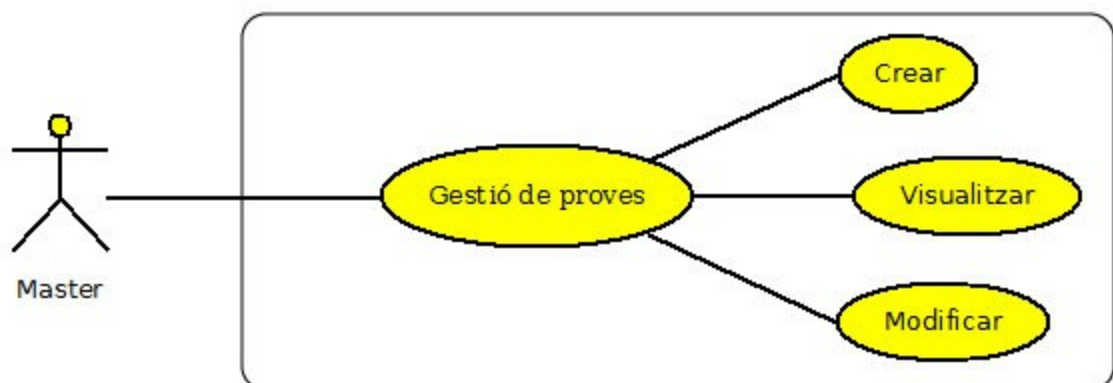
A continuació es mostraran tots els diagrames i fitxes de cas d'ús de la part web

#### 8.1.3.1 Administració de jocs



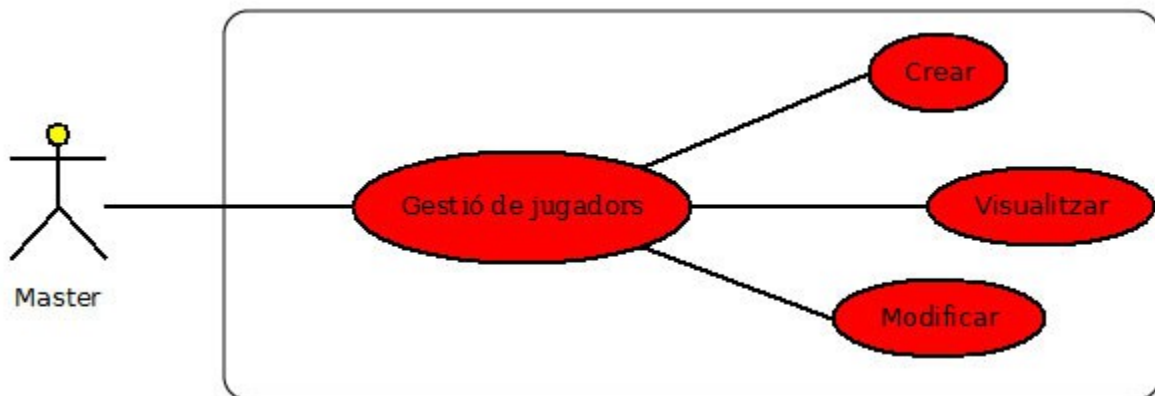
Gestionar Jocs	
Versió	Usuari
Actors	Màster
Flux principal	<ol style="list-style-type: none"> <li>1 S'obre la pantalla inicial del màster</li> <li>2 Obrir pantalla d'alta (clic a l'opció "Crear Joc")               <ol style="list-style-type: none"> <li>2.1 Entrar dades</li> <li>2.2 Guardar</li> </ol> </li> <li>3 A la pantalla inicial del màster es veuen tots els jocs generats pel màster (Visualitzar)</li> <li>4 Modificar dades d'un joc (clic sobre el nom del joc)               <ol style="list-style-type: none"> <li>4.1 Si modificar                   <ol style="list-style-type: none"> <li>4.1.1 Entrar dades</li> <li>4.1.2 Guardar</li> </ol> </li> </ol> </li> </ol>
Flux alternatiu	2.1 i 4.1.1 - Si les dades són incorrectes o hi ha camps obligatoris buits o incorrectes, mostra un missatge d'error detallat.

### 8.1.3.2 Administració de proves



Gestionar Proves	
Versió	Usuari
Actors	Màster
Flux principal	<ol style="list-style-type: none"> <li><b>1</b> S'obre la pantalla inicial del màster</li> <li><b>2</b> Obrir pantalla d'alta (clic a l'opció “Afegir proves al joc creat” i seleccionant el joc) <ol style="list-style-type: none"> <li><b>2.1</b> Entrar dades</li> <li><b>2.2</b> Guardar</li> </ol> </li> <li><b>3</b> Obrir la pantalla de visualització de proves (clic al nom del joc del que en volem veure les proves)</li> <li><b>4</b> Obrir el joc (clicant sobre el nom del joc), seleccionar la prova (clicant sobre el nom de la prova) <ol style="list-style-type: none"> <li><b>4.1</b> Si modificar <ol style="list-style-type: none"> <li><b>4.1.1</b> Entrar dades</li> <li><b>4.1.2</b> Guardar</li> </ol> </li> </ol> </li> </ol>
Flux alternatiu	2.1i 4.1.1. Si les dades són incorrectes o hi ha camps obligatoris buits o incorrectes, mostra un missatge d'error detallat.

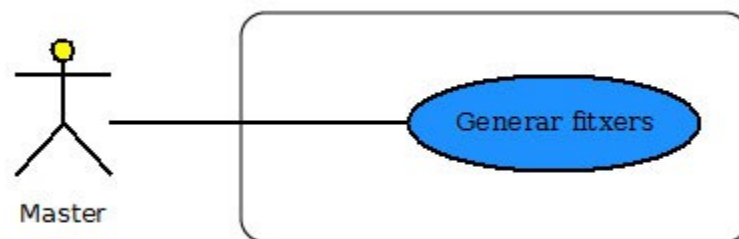
### 8.1.3.3 Administració de jugadors



Gestionar Jugadors	
Versió	Usuari
Actors	Màster
Flux principal	<ol style="list-style-type: none"> <li><b>1.</b> S'obre la pantalla inicial del màster</li> <li><b>2.</b> Obrir pantalla d'alta (clic a “Donar d'alta jugadors”) <ol style="list-style-type: none"> <li><b>1.</b> Entrar dades</li> <li><b>2.</b> Guardar</li> </ol> </li> <li><b>3.</b> A la pantalla inicial del màster es veuen tots els jugadors generats</li> </ol>

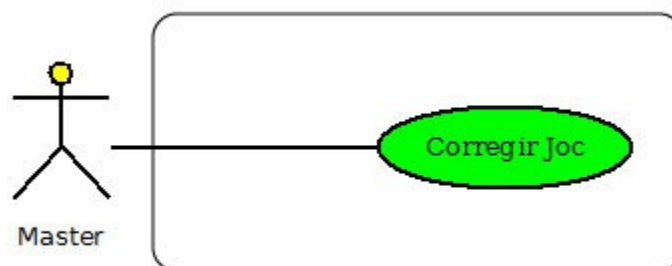
	pel màster (Visualitzar) <b>4. Obrir el jugador (clicant sobre el nom del jugador)</b> <b>1. Si modificar</b> <b>1. Entrar dades</b> <b>2. Guardar</b>
Flux alternatiu	2.1i 4.1.1. Si les dades són incorrectes o hi ha camps obligatoris buits o incorrectes, mostra un missatge d'error detallat.

#### 8.1.3.4 Generar fitxers



Generar Fitxers	
Versió	Usuari
Actors	Màster
Flux principal	<b>1</b> Clicar sobre l'opció “Generar fitxers del joc”, seleccionant el joc del que es vol generar els fitxers <b>1.1</b> Es generen tants arxius .xml com jugadors/equips hi ha, amb les dades del joc.
Flux alternatiu	cap

#### 8.1.3.5 Correcció i validació de jocs



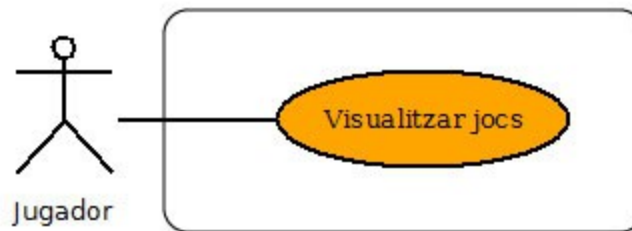


Corregir Joc	
Versió	Usuari
Actors	Màster
Flux principal	<ol style="list-style-type: none"> <li><b>1</b> Clicar sobre l'opció “Corregir joc”, seleccionant el joc del que se'n vol corregir les solucions. <ol style="list-style-type: none"> <li><b>1.1</b> Clicar sobre l'equip o jugador del qual se'n vol corregir les solucions <ol style="list-style-type: none"> <li><b>1.1.1</b> Validar les respostes</li> <li><b>1.1.2</b> Guardar <ol style="list-style-type: none"> <li><b>1.1.2.1</b> Es genera una puntuació final de cada equip o jugador</li> </ol> </li> </ol> </li> </ol> </li> </ol>
Flux alternatiu	cap

#### 8.1.4 Diagrames i fitxes de cas d'ús de la part Android

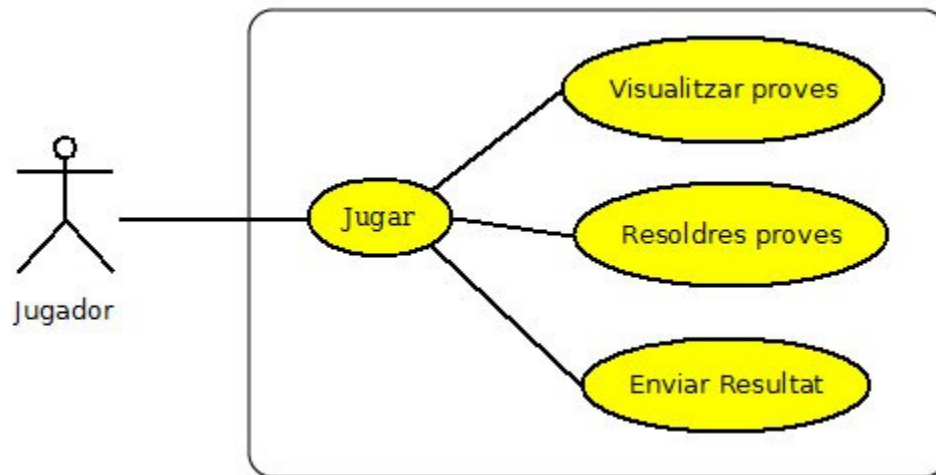
A continuació es mostraran tots els diagrames i fitxes de cas d'ús de la part Android

##### 8.1.4.1 Visualitzar Jocs



Visualitzar Jocs	
Versió	Usuari
Actors	Jugador
Flux principal	<ol style="list-style-type: none"> <li><b>1.</b> Entrant usuari i contrasenya correctes a la pantalla de login s'obre la pantalla amb els jocs en els que està donat d'alta el jugador.</li> </ol>
Flux alternatiu	cap

##### 8.1.4.2 Jugar



Jugar	
Versió	Usuari
Actors	Jugador
Flux principal	<ol style="list-style-type: none"> <li>1. Clicar sobre l'opció de "Jugar" del joc al que es vol jugar.</li> <li>2. Obrir la pantalla de visualització de la prova</li> <li>3. Obrir la pantalla de resolució de la prova               <ol style="list-style-type: none"> <li>1. Entrar resposta</li> <li>2. Guardar</li> <li>3. Si hi ha més proves, tornar al 2.</li> </ol> </li> <li>4. Clicar sobre l'opció "Enviar resultats"               <ol style="list-style-type: none"> <li>1. Es genera un fitxer XML amb els resultats de les proves del joc d'aquell jugador.</li> </ol> </li> </ol>
Flux alternatiu	cap

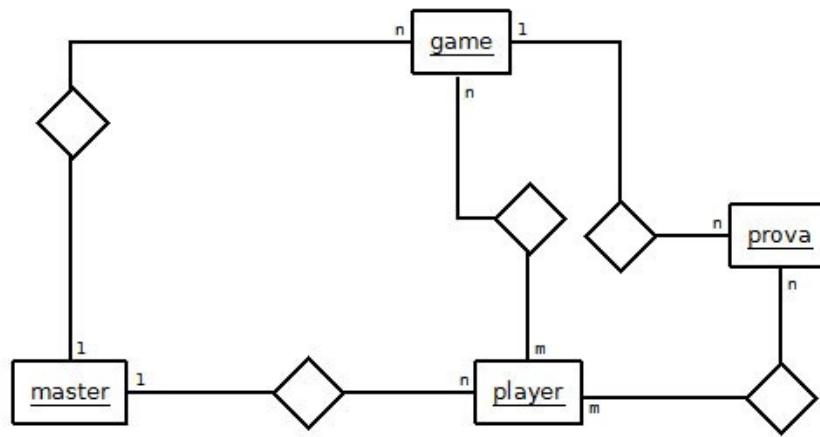
## 8.2 Disseny del sistema

### 8.2.1 Estructura de la Base de dades

Abans de començar a programar, es va optar per a realitzar el model de la base de dades, per tal de començar per quelcom bàsic i anar ampliant segons les necessitats definint, d'aquesta manera, l'estructura de les taules. Es va optar per utilitzar el model Entitat-Relació.

#### 8.2.1.1 Model entitat-relació

Seguidament, es mostra el model Entitat-Relació de l'aplicació:



### 8.2.1.2 Model Relacional

Després de realitzar el model entitat-relació, s'ha creat un model relacional per tal de visualitzar el disseny lògic.

Detallem totes les taules obtingudes de l'aplicació. La nomenclatura que s'utilitzarà serà:

- ♦ Els camps principals apareixen subratllats amb una línia contínua
- ♦ Els camps de tipus clau forana, apareixeran en *cursiva*

Així doncs, el model relacional serà el següent:

**GAME**(idGame, nom, loc, img, inici, *id\_màster*, *dataCreacio*, estat)

**PROVA**(id\_prova, nom, address, enunciat, punts, tipusProva, abc1, abc2, abc3, abc4, resposta\_ok, oberta1, oberta2, oberta3, oberta4, *nom\_joc*, *id\_màster*)

**MASTER**(idmaster, nom, cog1, cog2, dnaix, nick, passwd, mail, avatar, datacrea)

**PLAYER**(idplayer, nick, passwd, nom, cog1, cog2, mail, dnaix, dataCreacio, *id\_creador*, avatar)

Al generar dues relacions n:m (molts a molts), es creen dues taules auxiliars per emmagatzemar-hi dades (els anomenats atributs de la relació), que són les següents:

**GYMKHANA**(idGame, idPlayer, fitxerInicial, fitxerFinal, puntuacio)

**JUGADORPROVA**(id\_player, id\_prova, resultat, punts)

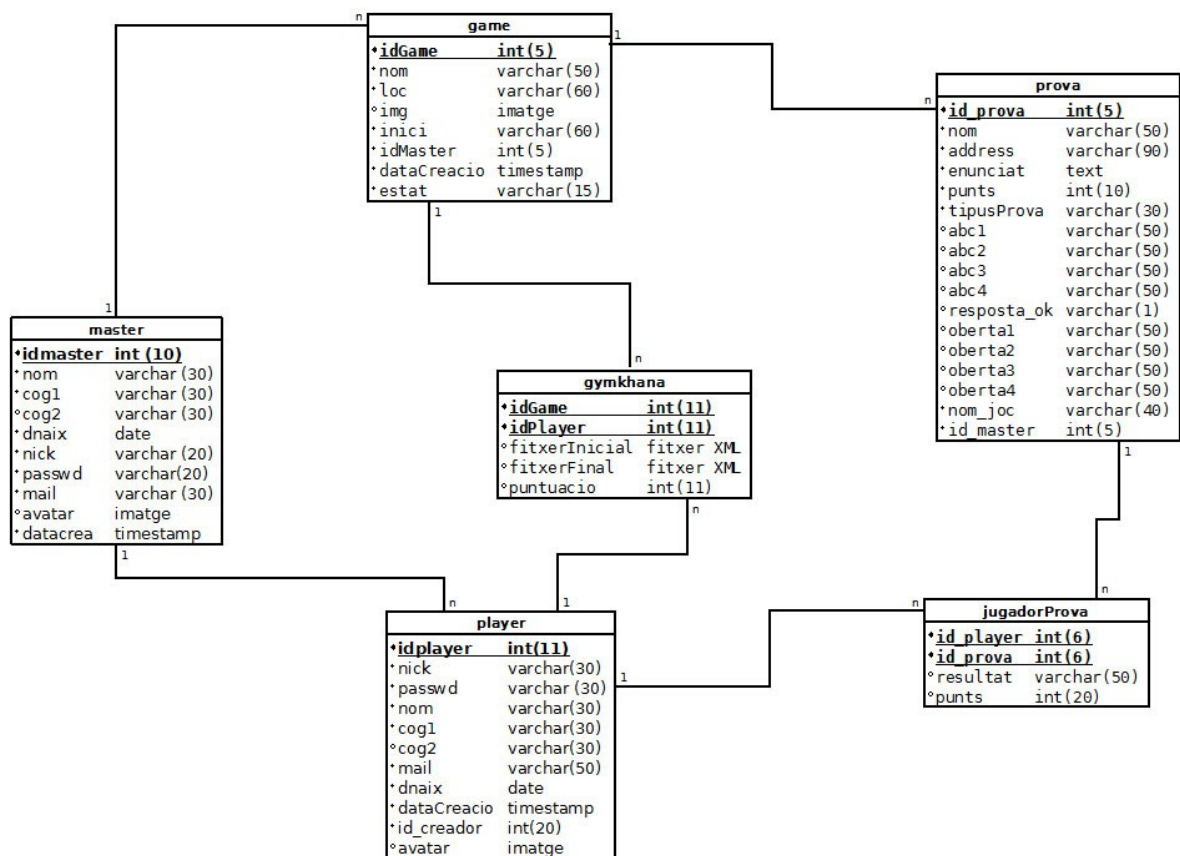
### 8.2.1.3 Explicació de les taules a la base de dades

Un cop mostrades les taules i les relacions entre elles, a continuació fem una breu explicació de cada taula de l'aplicació:

- ♦ **JOC:** En aquesta taula hi apareix tota la informació dels diferents jocs donats d'alta per un determinat màster.
- ♦ **PROVA:** Aquesta taula emmagatzema totes les dades de les proves relacionades amb un joc concret i realitzades per un màster.
- ♦ **MASTER:** En aquesta taula hi consten tots els màsters preparats per generar un joc.
- ♦ **JUGADOR:** Aquesta taula emmagatzema tots els jugadors donats d'alta al sistema, creats per un màster, i amb joc/s assignat/s.
- ♦ **GYMKHANA:** Aquesta taula és auxiliar i emmagatzemarà els atributs de la relació entre player i game, bàsicament els fitxers XML que es generin.
- ♦ **JUGADORPROVA:** Aquesta taula és auxiliar i emmagatzemarà els atributs de la relació entre una prova feta per un determinat jugador.

### 8.2.2 Diagrama de classes

A la figura següent es mostra el diagrama de classes de l'aplicació amb els atributs de cada classe:



## 8.2.3 Explicació classes

### 8.2.3.1 MainActivity.java

Aquesta classe representa l'Activitat principal, és a dir, és la pantalla inicial. MainActivity.java controla el comportament dels ítems que formen part de l'activitat principal. Els ítems estan distribuïts per l'activitat tal i com marca el fitxer activity\_main.xml, lligat a la classe de la que parlem. Aquest fitxer conté els elements que es veuen a la pantalla i atributs d'aquests elements, que farà servir la classe per a controlar l'activitat principal.

Com a atributs té els components visuals que es poden trobar a la pantalla i que l'usuari hi podrà interactuar.

- ♦ **EditText username:** és un camp del tipus EditText on l'usuari, per a voler-se loguejar, hi entrarà el nom d'usuari.
- ♦ **EditText password:** també és un altre camp del tipus EditText, i emmagatzemarà la contrasenya que entri l'usuari.
- ♦ **Button bgnLogin:** El botó per enviar les dades i que activarà una nova activitat.

Els mètodes principals de la classe MainActivity.java serien:

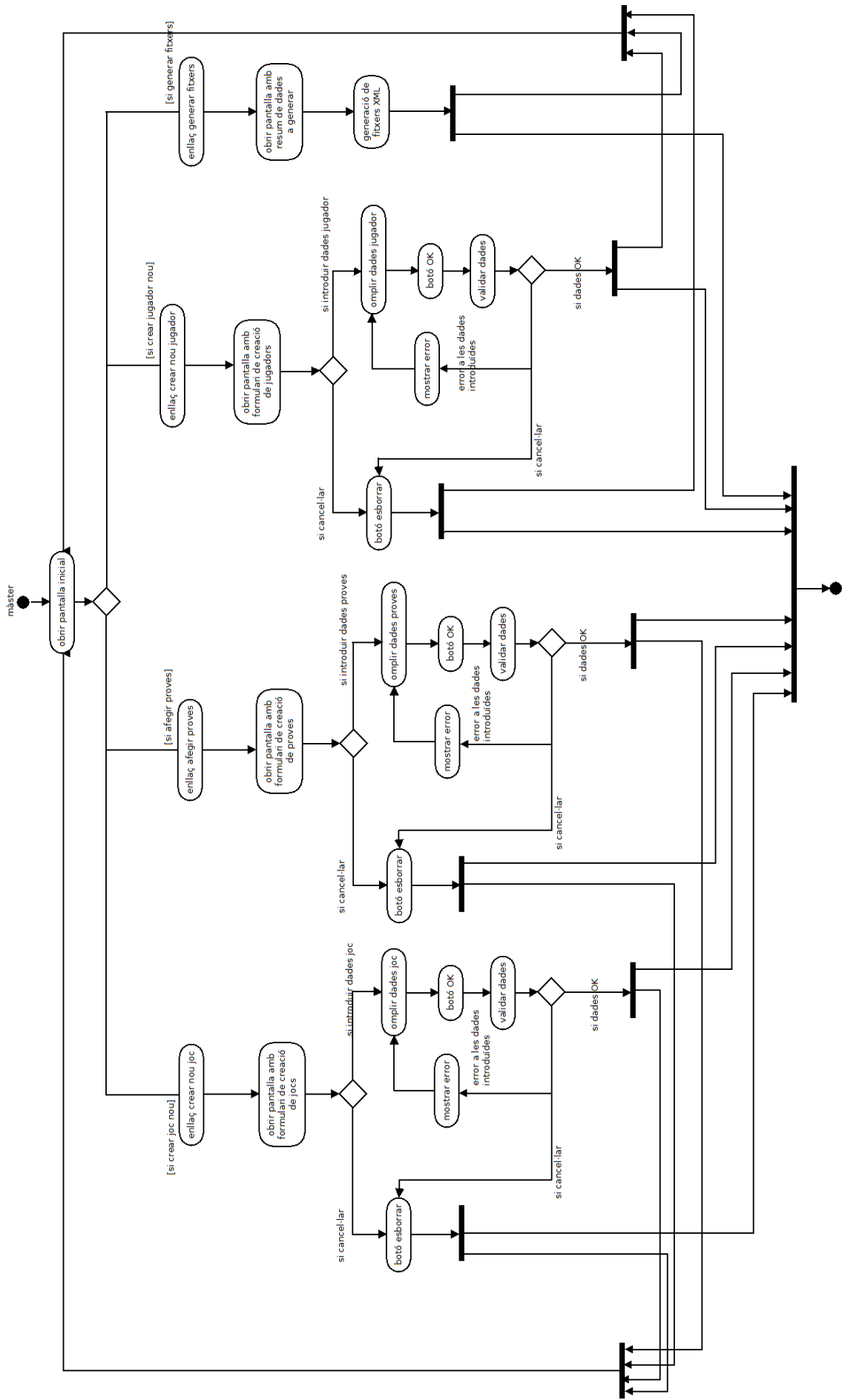
- ♦ **onCreate()**: és cridada quan l'activitat és creada
- ♦ **onCreateOptionsMenu (Menu menu)**: afegeix elements al menú de la barra superior (anomenada actionBar).
- ♦ **OnOptionsItemSelected (MenuItem item)**: aquest mètode es crida quan l'usuari clica a un ítem del menú d'opcions de la part superior de la pantalla, per oferir les diferents opcions.

### **8.2.4 Diagrames d'activitat**

En aquest apartat es mostraran els diagrames d'activitat que es volen destacar.

#### **8.2.4.1 Pantalla d'inici del màster**

A continuació es mostra el diagrama de la pantalla d'inici del màster, amb totes les opcions des les que disposa a l'inici. S'ha escollit aquesta pantalla, ja que és una de les que dóna més opcions diferents.





## 9 Implementació i proves

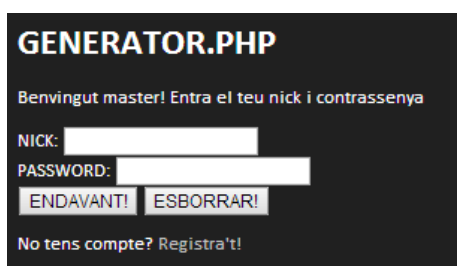
En aquest capítol s'exposarà per una banda la interfície d'usuari i per altra la implementació dels algoritmes i processos del projecte. Es detallarà la implementació de les classes, es realitzarà una descripció del funcionament. Cal dir que no es detallaran totes les implementacions que s'han realitzat, només les que es creuen més interessants en quant a implementació.

### 9.1 Interfície d'usuari

En aquest apartat es mostren els diferents tipus de pantalles de les aplicacions que pot utilitzar l'usuari, tant el màster com l'usuari jugador.

#### 9.1.1.1 Pantalla d'identificació web

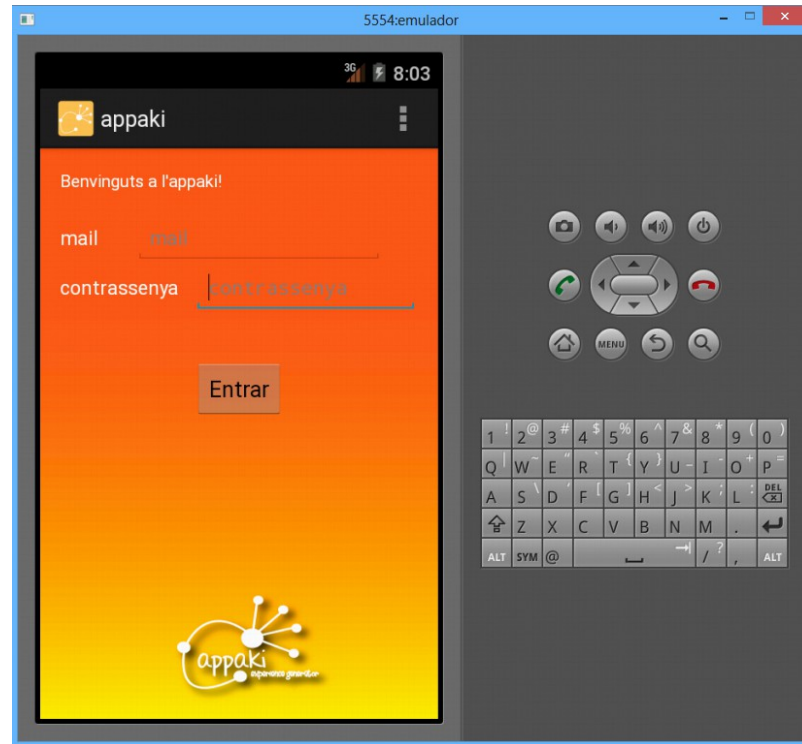
La pantalla d'identificació correspon a la pantalla d'autenticació dels usuaris per tal d'accedir a l'aplicació web.



En aquesta pantalla s'ha de posar el nom d'usuari i la contrassenya en els camps de text, i prémer el botó Endavant per accedir a l'aplicació. Si aquests són incorrectes, l'aplicació no deixarà avançar, i demanarà que, o bé ho tornin a intentar, o bé que es registrin al sistema.

#### 9.1.1.2 Pantalla d'identificació Android

La pantalla d'identificació correspon a la pantalla d'autenticació dels usuaris per tal d'accedir a l'aplicació Android.



En aquesta pantalla s'ha de posar el nom d'usuari i la contrassenya en els camps de text, i prémer el botó Entrar per accedir a l'aplicació. Si aquests són incorrectes, l'aplicació no deixarà avançar.

### 9.1.1.3 Pantalla d'inici web

La pantalla d'inici és la primera pantalla que apareix a l'entrar a l'aplicació després d'haver-se identificat correctament:

**PAGE2.PHP**

USUARI: taylor

Benvingut taylor!

Jocs creats:

ID	Nom	Nombre de proves	Localització	Data Creació	Estat	Correcció
26	Chronos	5	Girona	2014-08-16 17:52:29	obert	pendent finalitzar
20	Les Salines	2	Gerri	2014-08-10 15:41:48	tancat	corregir

Total de jocs creats: 2

Afegir proves al Joc creat

Chronos Quantes proves vols afegir?:  Afegir Proves!

- Generar fitxers del joc: (quan ja estigui preparat per jugar)

Chronos Generar fitxers!

Jugadors creats:

ID	Nick	Nom	Cognom	Cognom2	Mail	Data naix	Data creació	ID creador	avatar	Jocs
30	jacko	Michael	Joseph	Jackson	info@michaeljackson.com	1958-08-29	2014-09-01 05:51:43	1	IMG_0001.JPG	<ul style="list-style-type: none"> <li>Colors</li> <li>Chronos</li> </ul>

Total de jugadors creats: 1

- Crear un nou joc
- Crear un nou jugador
- Tancar la Sessió

En aquesta pantalla es poden observar diferents elements que formen part de la vista principal. Primer de tot, a la part superior, es pot observar la identificació de l'usuari actual, i un missatge de benvinguda. Aquesta pantalla pretén ser un resum de tot allò que ha creat el màster, i tot allò que pot crear/modificar. Per tant, el que es pot veure és, primerament, el resum dels jocs que ha creat el màster, detallats amb els atributs més importants. D'aquesta manera la informació es veu de seguida sense haver de fer altres clics/consultes.

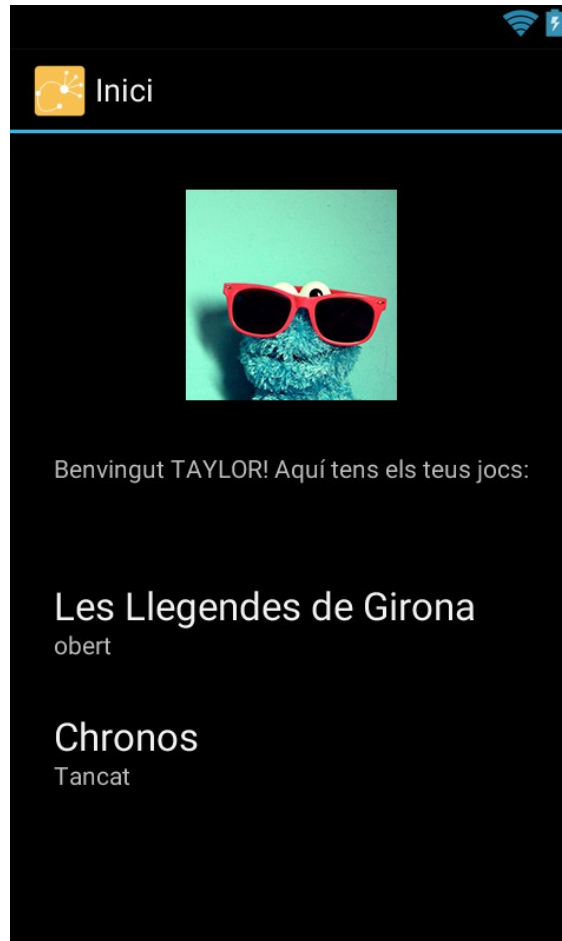
Just a sota dels jocs creats pel màster, es poden veure dos sub-operacions que es poden fer referents als jocs. La primera és afegir proves per, un cop creat el joc, poder-hi anar sumant proves. Seguidament, es troba l'operació de generar fitxer per, un cop generat el joc, donats d'alta els jugadors i creades les proves, a poder generar els diferents fitxers XML per a cada usuari diferent.

Seguint l'ordre descendent, s'hi pot trobar una altra taula-resum, però en aquest cas és dels jugadors donats d'alta per aquest màster. De la mateixa manera trobem alguns dels atributs més importants, com ara l'ID, el nick, el nom complet o bé els jocs als quals està donat d'alta.

Finalment, a la part inferior es pot observar un menú d'usuari que consta dels ítems: crear joc i crear jugador, que serviran per donar d'alta un joc i un jugador al sistema i, finalment, l'opció de tancar la sessió. Si aquesta opció és clicada, es retorna a la pàgina inicial de login.

#### **9.1.1.4 Pantalla d'inici Android**

La pantalla d'inici és la primera pantalla que apareix a l'entrar a l'aplicació després d'haver-se identificat l'usuari correctament. Es tracta de la pantalla amb l'avatar del jugador i els jocs els quals està donat d'alta, així com l'estat dels jocs:



#### 9.1.1.5 Pantalla de generació de jugadors

Un cop l'usuari màster prem “crear jugador”, s'obre la següent pantalla de figura següent ens demanen les dades del jugador, per tal d'emmagatzemar-lo, en aquest cas, a la base de dades.

Es pot observar com es demanen les dades bàsiques d'un jugador: nick, contrasenya, nom, cognoms, correu electrònic, data de naixement i avatar (aquest últim és opcional). Un cop

introduïdes aquestes dades, ens demanen que seleccionem a quins jocs volem donar d'alta el jugador, mitjançant una llista multi-selecció (es pot seleccionar més d'un joc). Un cop clicat sobre l'OK (botó) se'ns redirigirà a una pantalla on es resumeixen les dades entrades al sistema, que serà la pantalla de comprovació de les dades.

#### 9.1.1.6 Pantalla de generació de jocs

Quan l'usuari màster desitja crear un nou joc, clica a sobre l'ítem del menú “Crear joc nou” i apareix aquesta pantalla on es demanen les dades bàsiques per a la creació d'un nou joc:

Segons s'observa, es demana el nom del joc, la localització general (tipus municipi o lloc imprecís), la imatge que s'associarà al joc i, finalment, l'adreça de la base, que és el lloc a on comença el joc per a tots els jugadors. De la mateixa manera com al generar jocs, un cop s'entren les dades i es prem OK, es redirigeix al màster a una pantalla amb el resum de les dades entrades.

#### 9.1.1.7 Pantalla de generació de proves

Quan l'usuari vulgui crear una o més proves associades a un joc, a la part superior de la pantalla principal de l'usuari (punt 8.2.4.3) haurà de seleccionar el joc i el nombre de proves que s'hauran d'afegir, i després pressionar el botó “Afegir Proves”. Seguidament es mostra la part esmentada de la pantalla principal de l'usuari.

Un cop pressionat el botó “Afegir Proves”, es canvia la pantalla per la de generació de proves, que demana les dades tants cops com proves hi hagi per entrar. Aquesta pantalla de generació de proves és de l'estil que mostra la figura següent:

La part superior de la pantalla mostra les proves que ja consten en aquest joc (en el cas que n'hi hagi), per a situar a l'usuari màster en el joc, i que pugui saber quines proves ja han estat donades d'alta per l'usuari. El formulari demanarà per cada prova el nom, la localització (adreça) del punt de la prova, l'enunciat, la puntuació i el tipus de prova. Segons el tipus de prova que afegim (pregunta ABC, pregunta oberta), es desplegaran més opcions, tal i com mostraran les figures pròximes.

Si s'ha triat el tipus de prova com a **pregunta ABC** -que vol dir que serà una pregunta tipus test. El formulari demanarà que entrem les quatre opcions que se li donaran a l'usuari i quina és la solució.

Si s'ha triat el tipus de prova com a **pregunta oberta**, s'obriran uns altres camps que demanaran quatre possibles respostes correctes, útil per a poder fer bé la comprovació i correcció correctament.

Respostes

Degut a que és una pregunta oberta, contemplem la possibilitat de, tot i que és una sola resposta, ens l'entrim de maneres diferents. Escriviu aquí les possibles formes de la resposta. (La comprovació de majúscules o minúscules obvieu-la).

Resposta Correcta 1:

Resposta Correcta 2:

Resposta Correcta 3:

Resposta Correcta 4:

#### 9.1.1.8 Pantalla de generació de rutes

Un cop el joc estigui totalment preparat per a posar-se en marxa, que els jugadors estiguin tots assignats, i que les proves estiguin entrades, es pot generar els fitxers. Generar fitxers vol dir que es generarà un XML amb les dades del joc per a cada un dels usuaris, per a que no tinguin la mateixa ruta, sinó que tinguin un recorregut diferent. Per a fer-ho, des de la pantalla principal del màster, se selecciona un joc i seguidament es clica sobre “Generar fitxers”:

• Generar fitxers del joc: (quan ja estigui preparat per jugar)

Projecte Final de Carrera ▼

Projecte Final de Carrera

Festa Major de Gerri

Under The Dome

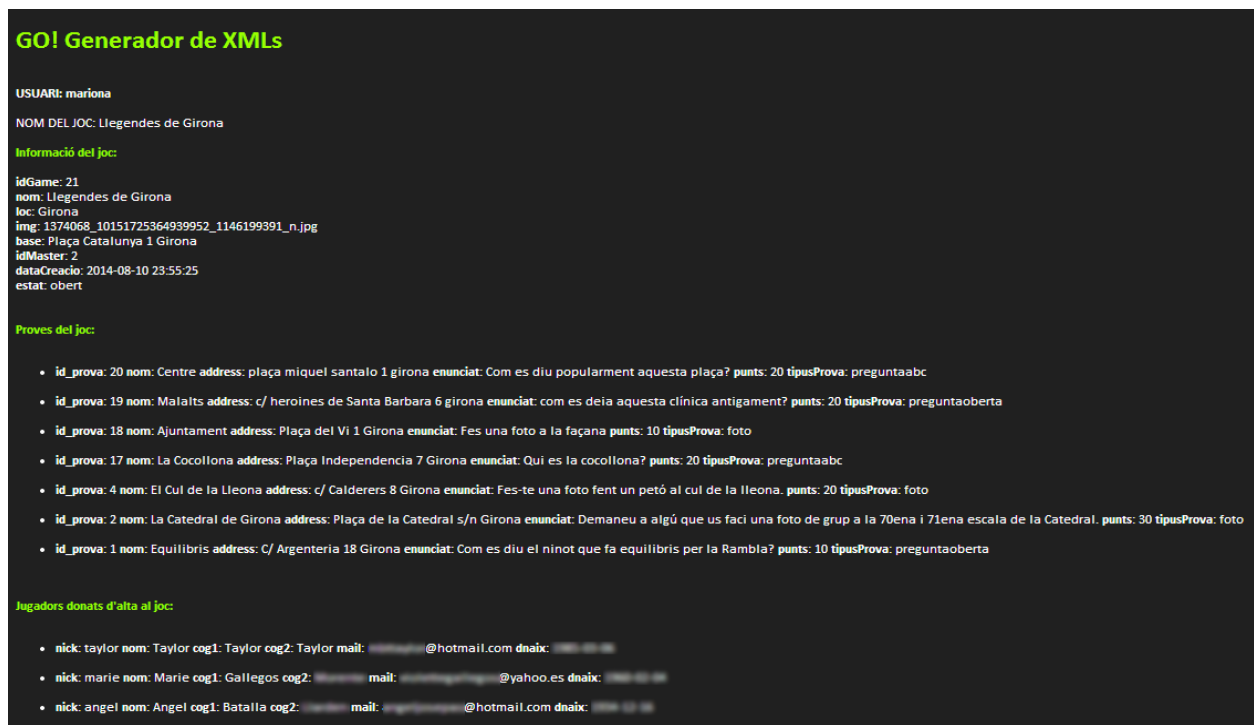
Llegendes de Girona

Gerri de la Sal

Generar fitxers!

Cognom Cognom2

Un cop clicat, passem a una pantalla de resum de dades, per tal de veure tota la informació que passarà a ser generada als fitxers que definiran el joc:



Un cop comprovada la correctesa de les dades, a la part inferior es pot fer clic sobre “GENERAR ELS FITXERS DEL JOC”, i a partir d'aquí el sistema calcularà una ruta diferent per a cada usuari, generant amb aquesta ruta i les dades del joc un fitxer XML que contindrà tot el joc d'un jugador concret. Aquest fitxer es guardarà al servidor i al mateix temps també hi posarem la ruta per accedir-hi des de la base de dades. És a dir, a la taula gymkhana de la base de dades s'afegirà, al registre amb clau idJoc, idJugador la ruta del fitxer XML del servidor, per a poder-se descarregar.

#### 9.1.1.9 Pantalla d'inici de joc (Android)

L'usuari (jugador) de l'aplicació d'Android, un cop ha passat la pantalla de login amb èxit i ha seleccionat el joc corresponent (sempre i quan l'estat del joc sigui obert), es troba amb la pantalla d'inici del joc corresponent, que té aquesta forma:



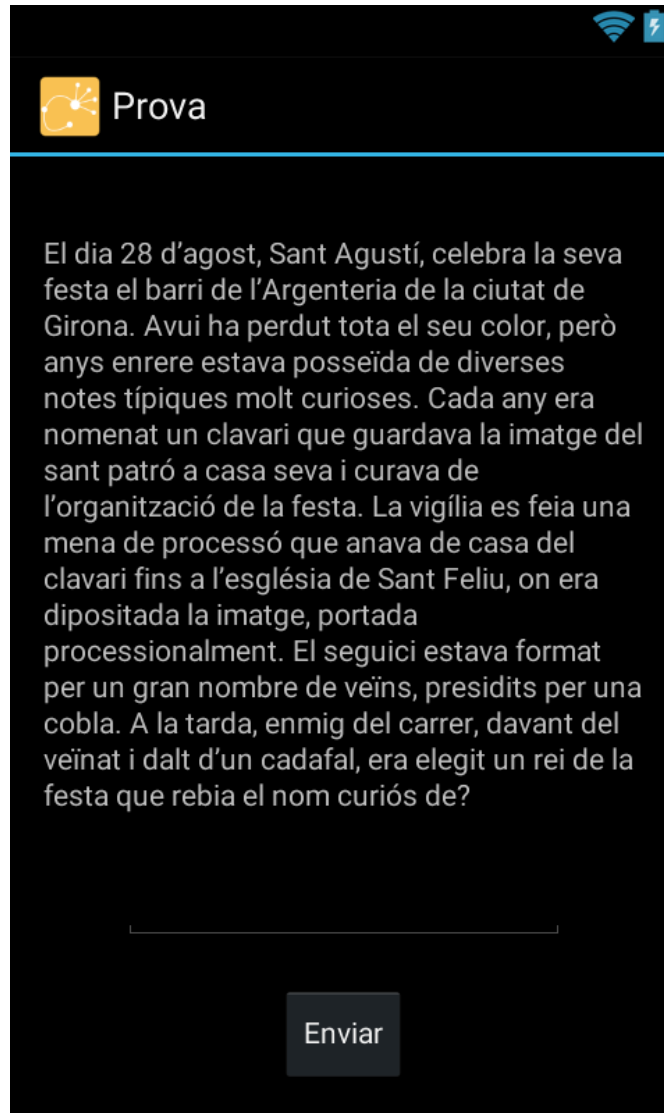


Com es pot observar, s'hi pot veure la imatge descriptiva o logo que ens han entrat a la base de dades, i una breu descripció o sinopsis de la temàtica del joc. També podem observar un botó que portarà a la primera pista de la primera prova del joc, que només estarà disponible quan l'estat del joc sigui OBERT, per tal de controlar els accessos no desitjats.

#### 9.1.1.10 Pantalla de prova (Android)

Un cop l'usuari s'ha aproximat a la zona de la prova, l'usuari podrà clicar el botó GO! (des de la pantalla de la localització de la prova -mapa) per a poder accedir als continguts d'aquella prova, és a dir enunciat i possibles respostes.

La pantalla de la prova l'hem preparada d'aquesta manera molt senzilla. Primerament surt l'enunciat i seguidament el lloc on contestar la pregunta i el botó d'enviar-la, és una interfície molt simple:



## 9.2 Implementació

### 9.2.1 Donar d'alta a usuaris

Per a la gestió d'usuaris primer de tot s'ha implementat un formulari molt bàsic en html per tal que s'entrin les dades que es necessiten per donar d'alta a l'usuari al sistema:

```
<form name="formulari_joc" action="resum_jugador.php" method="POST">

    <!--el formulari demanara nom-->
    <LABEL for="nick"> NICK: </LABEL>
    <!--el guardarem sota la variable nick-->
    <INPUT type="text" name="nick"><br>

    <!--el formulari demanara contrasenya-->
    <LABEL for="passwd"> PASSWORD: </LABEL>
    <!--el guardarem sota la variable pass-->
    <INPUT type="password" name="passwd"><br>

    <!--el formulari demanara nom-->
    <LABEL for="nom"> NOM: </LABEL>
```

```

<!--el guardarem sota la variable nom-->
<INPUT type="text" name="nom"><br>

<!--el formulari demanara cognom1-->
<LABEL for="cog1"> PRIMER COGNOM: </LABEL>
<!--el guardarem sota la variable cog1-->
<INPUT type="text" name="cog1"><br>

<!--el formulari demanara cognom2-->
<LABEL for="cog2"> SEGON COGNOM: </LABEL>
<!--el guardarem sota la variable cog2-->
<INPUT type="text" name="cog2"><br>

<!--el formulari demanara mail-->
<LABEL for="mail"> CORREU ELECTRONIC: </LABEL>
<!--el guardarem sota la variable mail-->
<INPUT type="email" name="mail"><br>

<!--el formulari demanara data de naixement-->
<LABEL for="dnaix"> DATA DE NAIXEMENT: </LABEL>

<!--el guardarem sota la variable dnaix-->
<INPUT type="date" name="dnaix"><br>
<!--el formulari demanara avatar del jugador-->
<LABEL for="avatar"> AVATAR: </LABEL>
<!--el guardarem sota la variable avat-->
<INPUT type="file" name="avatar" size=200 ><br>

<!--el formulari demanara a quins jocs se li ha de donar l'alta-->
<LABEL for="jocs"> JOCS ALS QUE SE LI HA DE DONAR ALTA: </LABEL>
<?php
    *llistarem els jocs*/
$result = mysql_query("SELECT nom FROM game ORDER BY dataCreacio DESC;") or
die(mysql_error());

    echo "<select multiple name=\"jocsTotals[]\" id=\"jocsTotals\">";
while($row = mysql_fetch_assoc($result))
{
    foreach($row as $cname => $cvalue)
    {
        echo "<option value=\"";
        echo "$cvalue";
        echo "\">";
        echo "$cvalue";
        echo "</option>";
    }
}
echo "</select>";
?>

<!--enviem dades!-->
<INPUT type="submit" value="OK!"><!--acceptar-->
<!--reestablim formulari-->
<INPUT type="reset" value="ESBORRAR!"><!--reset-->

```

Veiem com, a més a més també hi ha una llista desplegable on surten tots els jocs per tal que es puguin seleccionar (és de múltiple selecció) i assignar-los d'aquesta manera al jugador. Un cop aquest formulari és emplenat i l'usuari màster ha fet clic sobre el botó "OK", es passa a captar les dades del post del formulari per a entrar-les a la base de dades:

### 9.2.2 Connexió php amb MySQL

Seguirem amb el cas del punt anterior, el de donar d'alta un nou usuari a la base de dades, primer de tot, haurem d'establir una connexió amb la base de dades, en el cas que la connexió funcioni, retorna un identificador de l'enllaç amb el sistema de base de dades.

```
$connexio=mysql_connect("$host", "$user", "$pass") or die ("NO S'HA POGUT ESTABLIR CONNEXIÓ AMB LA BBDD");
```

Un cop tenim la connexió, hem d'indicar-li al sistema la base de dades a la que ens volem connectar:

```
mysql_select_db("Sql319769_4") or die ("NO S'HA POGUT SELECCIONAR BDD");
```

Primer de tot s'haurà de comprovar que el jugador no existeixi ja a la nostra base de dades, per tant, el que farem és una consulta sql que miri si el nombre de registres de la base de dades el nick (un dels camps claus) del qual coincideixi amb el que hem entrat és major que 0. Si és major que 0 això voldrà dir que el nom ja existeix i haurem d'avisar a l'usuari.

Primer guardem el nick que ens han entrat anteriorment en una variable i realitzem la pertinent consulta:

```
$nick = $_POST['nick'];

$total=mysql_num_rows(mysql_query("SELECT * FROM player WHERE nom='$nick'"));
$registres_bdd=mysql_num_rows(mysql_query("SELECT * FROM player"));
echo "<br> la taula player te $registres_bdd registres";

if($total!=0)
{
    echo "<br> El jugador ja consta a la base de dades!";
    echo "<br> <a href='generateplayers.php'>Tornar-ho a intentar</a>";
}

else{
```

Si no consta a la base de dades, haurem de donar-lo d'alta. També necessitarem la ID del màster que l'ha creat el jugador, perquè és una dada necessària per a afegir-los la base de dades

```
/*obtenim la ID del master actual*/
$idMaster=mysql_result(mysql_query($sql="SELECT `idmaster` FROM `master` WHERE `nick` = '$usuari'"),0);
```

Ara sí, ja podem crear un nou registre a la taula player (jugador) amb les dades del jugador que ens ha entrat, mitjançant aquesta consulta:

```
/*guardem la consulta SQL en una variable*/
$sql="INSERT INTO player(nick, passwd, nom, cog1, cog2, mail, avatar, dnaix, id_creador) VALUES('$nick','$passwd','$nom','$cog1','$cog2','$mail','$avatar','$dnaix', '$idMaster')" or die(mysql_error());
```

```
/*executem la consulta a la base de dades*/
mysql_query($sql);
```

Però aquí no acaba, recordem que a l'esquema de la base de dades, existeix una taula gymkhana on hi consten els jugadors emparellats amb els jocs als quals estan donats d'alta, per tant, necessitarem recuperar la ID d'aquesta nova alta que hem fet:

```
$idNouJugador=mysql_result(mysql_query("SELECT idplayer FROM player where nick
='$nick'"),0);
```

I crearem tants nous registres a la taula gymkhana com jocs diferents s'hagi de donar d'alta:

```
foreach ($_POST['jocsTotals'] as $selectedOption)
{
    /* $selectedOption sera el nom de cada un dels jocs*/
    //per tant, obtenim l'ID del joc a partir del nom:
    $idJoc=mysql_result(mysql_query("SELECT idGame FROM game where
nom='$selectedOption'"),0);

    $sql="INSERT INTO gymkhana(idGame, idPlayer)
VALUES('$idJoc','$idNouJugador')" or die(mysql_error());
    mysql_query($sql);
}

echo "<br> Perfecte! Jugador donat d'alta!!!";
```

### 9.2.3 Calcular la distància entre dos punts utilitzant l'API de google maps

Per a la geocodificació i el càlcul de distàncies s'ha utilitzat l'API de Google maps, tal i com s'ha exposat anteriorment. Per tant, el primer que hem posat al codi ha estat l'script amb la clau API que prèviament hem sol·licitat al web de Google Maps.

```
<script src="http://maps.google.com/maps?
file=api&v=2&key=AIzaSyCmW3LAa5DjzYxs1vhS0cmdy5PpyC4DeJw"
type="text/javascript"></script>
```

Es defineixen 4 variables locals que s'utilitzaran a les funcions. També es defineix, dins al tag <body> un event onload (al carregar-se), per a preparar l'API de Google Maps per les consultes i per calcular la distància entre dos punts donats:

```
function initialize()
{
    geocoder = new GClientGeocoder();
    gDir = new GDirections();
```

```

GEvent.addListener(gDir, "load", function() {
    var distanciaKilometres = gDir.getDistance().meters / 1000;
});
}

```

S'assigna a les variables locals geocoder i gDir, el geocoder de Google (GclientGeocoder) i l'API de direccions de Google (gDirections()). S'afegeix un listener a l'API de les direccions. Així, quan Google acabi de calcular la distància (entenem distància com la distància en “conducció”, és a dir, seguint les rutes, i no en línia recta) es duen a terme els pròxims passos. La distància és calculada mitjançant gDir.getDistance().meters en kilòmetres. Abans d'això s'hauran hagut de geocodificar les adreces que s'han entrat (una cadena de caràcters), ja que Google treballa amb elements LatLong per a fer els càlculs (coordenades GPS de latitud, longitud). Per a geocodificar les adreces utilitzarem la funció de l'API de Google Maps anomenada getLocation(), que converteix una adreça completament llegible a una parella de Latitud-Longitud, que emmagatzemarà també – a part de les coordenades latitud (lat) longitud (lon)- l'adreça (address) que s'ha entrat.

```

geocoder.getLocation(adreça, function (response)
{
    if (!response || response.Status.code != 200)
    {
        alert("ERROR: No s'ha pogut geocodificar l'adreça");
    }
    else
    {
        prova1 = {
            lat: response.Placemark[0].Point.coordinates[1],
            lon: response.Placemark[0].Point.coordinates[0],
            address: response.Placemark[0].address};
    }
}

```

Això es farà per a cada prova entrada a l'hora de calcular les diferents rutes d'un mateix joc.

#### 9.2.4 Generació de n rutes

Tal i com s'ha mencionat al llarg de la memòria, el sistema crearà una ruta de proves diferent per a cada jugador/equip. És per això que s'ha hagut de programar un algoritme pel càlcul i comparació de rutes. La idea bàsica és emmagatzemar els punts (les adreces de les proves) en una taula, juntament amb l'ID de la prova, per tal de poder-la identificar a l'hora de realitzar la ruta, i treballar a partir d'aquesta taula original.

Realitzarem la consulta a la base de dades per a que retorni totes les proves del joc que es demana, i recorrem la taula de resultats per a anar afegint les proves (l'id i l'adreça de la prova) a la taula que hem creat (\$taula).

```
//obtenir el nombre de proves que te el joc:

    $nProves=mysql_num_rows(mysql_query($sql="SELECT      *      FROM
prova WHERE nom_joc='$nomJoc'"));
    $_SESSION['numProves']=$nProves;

    $result  =  mysql_query("SELECT  id_prova,  nom,  address,
enunciat, punts, tipusProva FROM prova WHERE nom_joc = '$nomJoc' ORDER
BY id_prova DESC;")or die(mysql_error());

    //construim una taula per guardar-hi les adrces
    $taula=array();

    //recorrem la taula de resultats
    while($row = mysql_fetch_assoc($result))
    {

        foreach($row as $cname => $cvalue)
        {
            if($cname=="id_prova")$id_prova=$cvalue; //guardem
la id de la prova per tal de fer la taula
            if($cname=="address")
            {
                //sera una taula de taules,
                //necessitarem guardar l'ID de la prova i
l'adreça que serà un string
                $taula[]=array($id_prova,$cvalue);// a cada
posicio hi haura una taula que conte el n°ID i l'adreça
            }
        }
    }
}
```

Un cop tenim aquesta taula de proves, on hi haurà totes les proves del joc actual i que conté en cada posició una taula de dos elements (id i adreça de la prova); s'haurà de generar tantes rutes com jugadors/equips. Per a obtenir n rutes diferents, caldrà generar-ne més quantitat per tal que les n rutes escollides no diferenciïn gaire entre elles en quant a cost (entengui's cost com a metres totals que separen una prova de l'altra seguint una ruta donada). És per això que s'ha generat 3n (essent n el nombre de jugadors/equips) rutes diferents, fent una còpia de la taula original i desordenant-ne els elements aleatòriament. Aquestes 3n rutes s'emmagatzemaran en una altra taula més gran, que anomenarem \$comb i que contindrà tant la referència a la ruta (és a dir la taula proves), com el cost d'aquesta ruta (pel que s'utilitzarà el codi de l'apartat 9.4), en kilòmetres.

Per tant, el primer que es farà serà crear la taula \$comb (la taula gran) i seguidament generarem les 3n còpies, que anirem afegint a la taula gran, tal i com mostra el codi següent:

```
$comb=array();
/*la taula contindra una altra taula amb [0] cost total i [1] taula
de ruta*/
for($i=0;$i<(3*$nJugadors);$i++)
{
    //copiem taula:
```

```

$copia=$taula;
shuffle($copia);

//hem de comprovar que la ruta no existeixi ja
while( estaRepe($comb, $copia))
{
    //desordenem taula
    shuffle($copia);
}
//mostrar_array($copia);
$comb[]=array(calcular_cost($copia),$copia); // a cada posicio
hi haura una taula que conte el el cost i la taula d'adreces/proves
}

//ara ordenarem la taula "gran" seguint l'ordre de cost, de petit a
gran
asort($comb);

```

Un cop s'ha emplenat la taula \$comb amb 3n elements (rutes), el que farà serà ordenar-les pel cost, i s'obtindrà 3n rutes diferents ordenades pel cost. A partir d'aquí, s'ha d'escollir el subconjunt de n elements d'aquesta taula. Com a criteri s'ha fet servir el de la mediana. És a dir s'agafaran els n elements centrals de la taula, que seran els n elements que s'utilitzaran com a rutes pels n jugadors/equips del joc.

```

/*un cop ordenat agafarem el subconjunt de n elements a partir de la meitat
de la taula, fent caure la meitat de N també allà mateix*/

//longitudTaula/2 sera la meitat, i començarem a agafar els n
recorreguts a partir de (longitudTaula/2)-(numJugadors/2)
//per tant, fem el recorregut:
$mediana=sizeOf($comb)/2;
//arrodonim cap avall:
$inici=round($mediana-( $nJugadors/2), 0, PHP_ROUND_HALF_DOWN);

```

### 9.2.5 Accés a la base de dades des d'Android (login d'usuaris a l'aplicació Android)

Per a poder controlar l'accés a l'aplicació s'ha iniciat aquesta amb una Activity de Login que el que fa és recollir les dades de nom d'usuari i contrasenya d'uns elements de text i, quan es prem el botó, el sistema fa la comprovació a la base de dades de si hi ha algun jugador amb aquest mail identificatiu i aquesta contrasenya.

Per a fer això, primerament s'ha creat una nova Activity , i s'han afegit els elements desitjats, en aquest cas dos entrades de texts, el botó, i un fons de pantalla, així com dues etiquetes de text per a saber què s'ha d'entrar. El layout de l'Activity el tenim a continuació:

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fons_login"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"

```



```

tools:context="com.taylorslittlethings.appaki.MainActivity" >

<!-- botó de OK -->

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="@string/welcome"
    android:textColor="#FFFFFF" />

<EditText
    android:id="@+id/email"
    android:layout_width="200dp"
    android:layout_height="40dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/textView1"
    android:layout_marginRight="28dp"
    android:layout_marginTop="17dp"
    android:ems="10"
    android:hint="@string/mail"
    android:textColor="#000000"
    android:textSize="15sp" />

<EditText
    android:id="@+id/pword"
    android:layout_width="200dp"
    android:layout_height="40dp"
    android:layout_alignLeft="@+id/button1"
    android:layout_alignParentRight="true"
    android:layout_alignStart="@+id/email"
    android:layout_below="@+id/textView2"
    android:ems="10"
    android:hint="@string/passwd"
    android:inputType="textPassword"
    android:textColor="#000000"
    android:textSize="15sp" >

    <requestFocus />
</EditText>

<TextView
    android:id="@+id/pworda"
    android:layout_width="200dp"
    android:layout_height="40dp"
    android:layout_alignBaseline="@+id/pword"
    android:layout_alignBottom="@+id/pword"
    android:layout_alignEnd="@+id/textView2"
    android:layout_alignRight="@+id/textView2"
    android:layout_alignLeft="@+id/textView2"
    android:text="@string/passwd"
    android:textColor="#FFFFFF"
    android:textSize="15sp" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="200dp"
    android:layout_height="40dp"
    android:layout_alignBaseline="@+id/email"
    android:layout_alignBottom="@+id/email"

```

```

    android:layout_alignLeft="@+id/textView1"
    android:layout_toLeftOf="@+id/textView"
    android:layout_toStartOf="@+id/textView"
    android:text="@string/mail"
    android:textColor="#FFFFFF"
    android:textSize="15sp" />

```

```

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/pworda"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="37dp"
    android:onClick="passarPrincipal"
    android:text="@string/ppal" />

```

```
</RelativeLayout>
```

Es pot comprovar que, tal i com s'ha mencionat anteriorment, es tracta d'un fitxer XML, que té la seva versió Java (MainActivity.java), que defineix el comportaments d'aquests components que s'han definit al XML. Bàsicament s'ha redefinit el mètode onCreate(), d'aquesta manera:

```

public void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    preferences = PreferenceManager.getDefaultSharedPreferences(this);
    Editor edit = preferences.edit();
    edit.putString("uNom", "guest");
    edit.putString("pWord", "guest");
    edit.putBoolean("loggedIn", false);
    edit.commit();
    etUsername = (EditText)findViewById(R.id.email);
    etPassword = (EditText)findViewById(R.id.pword);
    btnLogin = (Button)findViewById(R.id.button1);

    // afagem el botó definit a l'xml i l'assignem a una variable local del
    tipus button:

    Button launch = btnLogin;

    //l'action listener
    launch.setOnClickListener(new View.OnClickListener()
    {
        public void onClick(View viewParam)
        {
            String sUserName = etUsername.getText().toString();
            String sPassword = etPassword.getText().toString();
            ArrayList<NameValuePair> postParameters = new
ArrayList<NameValuePair>();
            postParameters.add(new BasicNameValuePair("username",
sUserName));
            postParameters.add(new BasicNameValuePair("password",
sPassword));

            String response = null;
            try {
                //this

```

```

        response =
CustomHttpClient.executeHttpPost("http://www.taylorsslittlethings.com/appaki/login.php", postParameters);

        String res=response.toString();

        res= res.replaceAll("\\s+", "");

        if(res.equals("1"))
        {
            String LoginMsg = "Entrant...!\nUsuari: "
                + sUserName + "\nPassword: " +
sPassword;

            Toast.makeText(MainActivity.this,
LoginMsg, Toast.LENGTH_LONG).show();

            Intent MenuIntent = new
Intent(MainActivity.this,MainActivity.class);/**
startActivity(MenuIntent);
        }
        else
        {
            Toast.makeText(MainActivity.this, "Invalid
Login", Toast.LENGTH_LONG).show();
        }
    }
    catch (Exception e)
    {

        etUsername.setText(e.toString());

    }
}
});
}

```

Bàsicament aquest mètode crea tota la interfície que s'ha definit a l'XML i gestiona tot el que passa quan l'usuari fa un clic al botó. Quan l'usuari clica, el text que s'ha entrat per a identificar-se es transforma a String i s'emmagatzema en una taula de parelles de valors. Seguidament es realitza la crida al codi PHP que farà la gestió de mirar la base de dades i si la contrasenya és correcta. Aquesta crida es fa amb aquest mètode: `executeHttpPost`. Per a que aquesta crida es pugui dur a terme, és necessari que a l'AndroidManifest.xml s'hi afegeixi això:

```
<uses-permission android:name="android.permission.INTERNET" />
```

L'script en php que hem penjat al nostre servidor és el següent, bàsicament es tracta d'una consulta SQL:

```

<?php

$un=$_POST['username'];
$pw=$_POST['password'];

//connectem a la base de dades
$user = 'lalala';
$pswd = 'lalala';
$db = 'lalala';
$conn = mysql_connect("xx.xxx.xxx.xx", $user, $pswd);
mysql_select_db($db, $conn)or die(mysql_error());

```

```

/*executem la query per cercar l'usuari (mail en aquest cas) i la
contrasenya dins la taula*/
$query = "SELECT * FROM player WHERE mail = '$un' AND passwd = '$pw'";

$result = mysql_query($query) or die("Unable to verify user because : " .
mysql_error());

//aquí es fa la verificació
if(mysql_num_rows($result) > 0) echo 1; // mostrem 1 si s'ha trobat un
registre i per tant el login es correcte
else echo 0; // mostrem 0 si no s'ha trobat cap registre, per tant el
login es incorrecte

mysql_close();//tanca la connexio
?>

```

Veiem que retorna 0 si no és correcte el login, i retorna 1 si el login és correcte. Si el login és correcte, deixa avançar cap a la pròxima activity.

### 9.2.6 Visualitzar els elements de la base de dades amb Android (visualització dels jocs del jugador)

De la mateixa manera que al punt anterior s'ha accedit a la base de dades mySQL mitjançant un script penjat al servidor, a l'hora de mostrar tots els jocs d'un jugador també s'ha realitzat d'aquesta manera. La diferència, però, és que aquest cop la resposta no és un “sí” o “no” sinó que retorna una sèrie d'informació que després es mostrarà a l'usuari. Funciona molt semblant, però aquest cop hi ha dades de retorn que han de ser processades o mostrades, per això s'ha preparat en l'activity una llista d'ítems per a que un cop la base de dades retorni els jocs (en aquest cas), cada ítem es posicioni com a un element de la llista i el mostri.

### 9.2.7 Descàrrega del fitxer XML a l'aplicació Android

Un cop hem tingut els fitxers XML generats per cada un dels jugadors, pujats al servidor, i amb la ruta a la base de dades, ha vingut el moment de baixar-se l'arxiu XML del servidor. S'ha fet servir un snippet per descarregar arxius des de la web, que el posem a continuació:

```

try {
//s'especifica l'origen de l'arxiu a descarregar:
URL url = new
URL("http://www.taylorslittlethings.com/appaki/test1.xml");

//establim la connexió amb el destí
URLConnection urlConnection = (URLConnection) url.openConnection();

//farem servir el mètode GET a l'hora de connectar-nos
//el mètode setdooutput és imprescindible per aquestes connexions
urlConnection.setRequestMethod("GET");
urlConnection.setDoOutput(true);

//per últim establim la connexió
urlConnection.connect();

```

```

//descarregarem aquest arxiu a l'arrel de la targeta SD:
File SDCardRoot = Environment.getExternalStorageDirectory();

//Ara es crea un objecte de tipus fitxer on hi descarregarem el nostre fitxer

File file = new File(SDCardRoot,"ejemplo.txt");

//utilitzarem un objecte del tipus fileoutputstream
//per escriure l'arxiu que descarreguem al nou
FileOutputStream fileOutput = new FileOutputStream(file);

//Llegim des de la URL
InputStream inputStream = urlConnection.getInputStream();

//obtenim la mida de l'arxiu i l'associem a un enter -per saber l'estat
int totalSize = urlConnection.getContentLength();
int downloadedSize = 0;

//creem buffer i una variable per anar emmagatzemant
byte[] buffer = new byte[1024];
int bufferLength = 0;

//recorrem el buffer per escriure l'arxiu final
while ( (bufferLength = inputStream.read(buffer)) > 0 )
{

fileOutput.write(buffer, 0, bufferLength);
downloadedSize += bufferLength;

}
//tanquem el fitxer
fileOutput.close();

//i gestionem els errors
}
catch (MalformedURLException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}
}

```

## 10 Implantació i resultats

A l'hora d'implementar el sistema s'ha tingut en compte dues lleis que afecten la majoria dels projectes informàtics: la LOPD i la LSSI.

### 10.1 Protecció de dades



#### **LOPD (Llei Orgànica de Protecció de Dades de Caràcter Personal)**

La Llei Orgànica 15/1999 de Protecció de Dades de Caràcter Personal (LOPD) és una llei orgànica que té com a objectiu garantir i protegir les dades personals, les llibertats públiques i els drets fonamentals de les persones físiques, i especialment la seva intimitat i privadesa personal i familiar. Fou aprovada a les Corts espanyoles el 13 de desembre de 1999. Aquesta llei es desenvolupa fonamentada en l'article 18 de la Constitució Espanyola de 1978, sobre el dret a la intimitat familiar i personal; i el secret de les comunicacions.

Les dades facilitades pels usuaris en l'aplicació són protegits d'un ús fraudulent i no es facilitarà cap de les dades a tercers.

#### **LSSI-CE (Llei de Serveis de la Societat de la Informació)**

La Llei de Serveis de la Societat de la Informació i del Comerç Electrònic 34/2002, també anomenada Llei del Comerç Electrònic, va entrar en vigor l'11 de juliol de 2002, per a regular els aspectes bàsics del comerç electrònic. És la llei que regula les comunitats electròniques, evita abusos i protegeix als consumidors, i en ella s'hi estableixen les obligacions, responsabilitats, infraccions i sancions de persones i empreses que tenen una pàgina web o que operen a través d'internet.

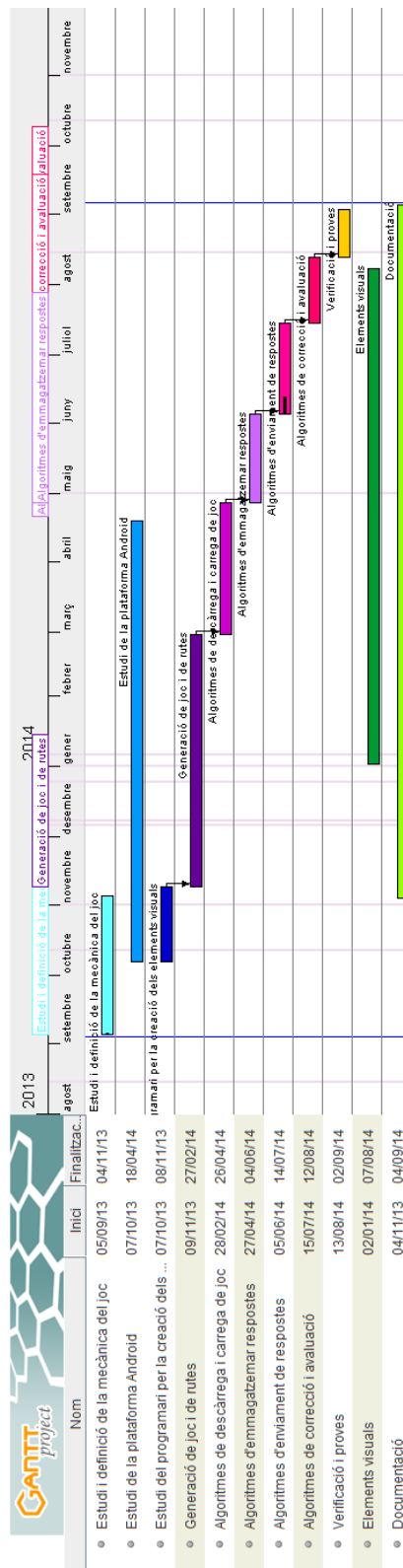
## 11 Conclusions

Un cop finalitzat el projecte, estic satisfeta de la feina realitzada i el gran esforç que he fet per assolir-ho, així com dels coneixements adquirits.

Un dels objectius principals de la realització d'aquest projecte era poder ser capaç d'aplicar tots els coneixements que se m'han donat a la carrera, començant una aplicació des de l'inici, passant per tots i cada un dels passos . Ha estat com donar-li sentit a tot el ventall de diagrames i conceptes que se'ns van ensenyar.

Aquest projecte m'ha permès creure una mica més en mi mateixa en aquesta àrea de gestió d'aplicacions, i adquirir l'experiència en la realització de projectes per a plataformes mòbils, que m'interessa molt de cares a un futur. M'ha sorprès la capacitat que he tret per a solucionar els problemes que se m'apareixien. Ha sigut una injecció de motivació extra molt forta ja que, malgrat l'any de circumstàncies personals dolentes, he estat capaç de mirar endavant i acabar aquest projecte tant important per a mi.

Respecte a la temporització, com he esmentat anteriorment, no s'ha pogut complir, ja que a part d'estar en el món laboral a jornada completa, he hagut de passar per circumstàncies personals molt difícils que m'han frenat en moltes ocasions. Seguidament, una aproximació del que ha sigut el diagrama de Gantt real que, com es pot observar, difereix de l'inicialment previst.





## 12 Treball futur

La realització d'aquest projecte ens ha dut a obtenir una aplicació bàsica, però que és una molt bona base per a incorporar-hi més ítems de forma relativament fàcil. Així doncs, com a possibles millores, es podria pensar primer de tot en millorar les interfícies d'usuari fent-les més elaborades i atractives, especialment per a l'aplicació Android.

Altres funcionalitats que seria interessant d'afegir en un futur podrien ser:

- ♦ Que es puguin fer equips a més d'usuaris individuals.
- ♦ Que les respostes puguin ser també de tipus vídeo curt o nota de veu, twitter.
- ♦ Dissenyar una manera més elaborada de donar les pistes.

## 13 Bibliografia

W3schools. 1999-2014. REFSNES DATA AS. 2012-2014 <http://www.w3schools.com>

Wikipedia, la enciclopedia libre. 2001-2014. Fundacion Wikimedia. 2011-2014.  
<http://es.wikipedia.org>

Php. 2001-2014. The PHP Group. 2011-2014. <http://www.php.net/>

StackOverflow. 2008-2014. Stack Exchange Network. 2011-2012.  
<http://stackoverflow.com>

Android Developers. 2010-2014  
<http://developer.android.com/intl/es/develop/index.html>

Basic4Android: Rapid App Development for Android. WYKEN SEAGRAVE (2013) -Kindle eBook